Simultaneous Dimensionality and Complexity Model Selection for Spectral Graph Clustering

by

Congyuan Yang

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

January, 2019

Abstract

Many real world applications involve the study of statistical inference on graphs. Our problem of interest is to cluster the vertices of a graph by identifying its underlying community structure. Among various vertex clustering approaches, spectral clustering is one of the most popular clustering methods because it is easy to implement while it often outperforms traditional clustering algorithms. However, there are two inherent model selection problems in spectral clustering, namely estimating the embedding dimension and estimating the number of clusters. Traditional model selection approaches determine the two model parameters successively, yet the consecutive procedure suffers from the intrinsic features of the framework such as subjectivity and accumulation of errors. This gives rise to the challenge of effective model selection for vertex clustering.

This thesis attempts to address the issue by establishing a novel model selection framework specifically for vertex clustering on graphs with stochastic block model. The first contribution of this thesis is a probabilistic model which

approximates the distribution of the extended spectral embedding of a graph. The model is constructed based on a theoretical result of asymptotic normality of the informative part of the embedding, and on a simulation result of limiting behavior of the redundant part of the embedding. The second contribution of this thesis is a simultaneous model selection framework. In contrast with the consecutive alternatives, this model selection procedure estimates embedding dimension and number of clusters simultaneously. Based on our proposed distributional model, a theorem on the consistency of the estimates of model parameters is stated and proven. The theorem provides statistical support for the validity of our method. Heuristic algorithms via the simultaneous model selection framework for vertex clustering are proposed, with good performance shown in the experiment on synthetic data. Finally, we demonstrate our methods on the real application of connectome analysis.

Primary Reader and Advisor: Dr. Carey E. Priebe

Secondary Reader: Dr. Trac D. Tran

Acknowledgments

First of all, I would like to express my deep gratitude to my advisor, Dr. Carey E. Priebe, for introducing me to the fantastic world of statistical learning, and for his insightful guidance along the road of my Ph.D. study. I am very grateful to him for being patient and supportive throughout my research. He always gives me the freedom to explore unknown paths and encourages me when I feel confused and frustrated in a hard time. His generous help shapes my research interests and makes my Ph.D. life memorable.

I am grateful to Dr. Trac D. Tran and Dr. Najim Dehak for serving in my thesis proposal and dissertation committees. Their valuable comments and suggestions greatly help me during my research. I am also thankful to Dr. René Vidal for the guidance of my first research project in Johns Hopkins. I have learned a lot from him on scientific research, writing and presentation. I would also like to thank Dr. Daniel Q. Naiman and Dr. Daniel Robinson for serving in my graduate board oral committee and being my mentors in the math courses.

ACKNOWLEDGMENTS

It has been an enjoyable experience to be part of the ECE department at Johns Hopkins University. I would like to thank Ms. Debra Race and Ms. Dana Walter-Shock for always being helpful with the administrative work. I also thank Dr. Shangsi Wang, Dr. Chong You and many other labmates who help and enlighten me through collaboration and discussion. I also thank many of my friends for giving me a delightful experience in Hopkins.

Finally, my deepest gratitude to my parents. I could not have been here without their unlimited love and support.

Dedication

This thesis is dedicated to my parents, Yongjing Lu and Jianmin Yang, for their eternal love and support.

Contents

CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

A vast number of real world applications involve the study of statistical inference on graphs. A mathematical graph encodes the complex relationships between objects in a network as edges between vertices. The analysis of such network is of ubiquity and importance in many fields ranging from sociology [56] and ecology [86] to political science [109] and neuroscience [13]. For example, graphs are used to capture the interactions among individuals in social network, or to model the connectivity structure between neurons and synapses in brain. Although the traditional graph theory has been studied for hundreds of years, the comparatively new notion of random graphs has received increasing attention over the past decades with the explosive growth of machine learning. This calls for the development of statistical techniques for uncovering the underlying properties of random graphs.

1.1 Vertex clustering based on stochastic block model

One of the most important tasks in the analysis of a complex graph is to identify its community structure. Specifically, the vertices in a same community usually share a common connectivity behavior, which is distinguishable from that of the vertices in other communities. In this sense, community detection is essentially a *vertex clustering* problem, in which the set of vertices is to be partitioned according to the underlying communities. In general, the traditional clustering task of unsupervised learning is to group a set of objects based on their similarities in some sense. Analogously, the vertex clustering problem can be defined as to divide the vertices of a graph into nonoverlapping groups (called clusters) in such a way that vertices in the same cluster are more similar to each other than to the vertices in other clusters.

Numerous heuristic methodologies have been proposed for vertex clustering, including divisive approaches by iteratively removing edges based on number of shortest paths [33, 77], methods of optimizing a function called "modularity" which evaluates the quality of a partition [7, 10, 76], and algorithms employing random walk to infer structural properties of networks [83, 92], to name a few. While heuristic methods are usually easy to implement and effective in specific scenarios, they lack the theoretical basis of consistency, namely

the proportion of misclassified vertices going to zero as graph size grows. In contrast, model-based approaches share the advantage of such theoretical support based on the assumption that the graph is generated from some parametric model. With a proper model, the parameterization makes it possible for us to utilize classical probabilistic and linear algebraic techniques to analyze the statistical properties of such graphs. The performance of subsequent inference task of vertex clustering can then be ensured directly following the theoretical results.

In the context of community detection or vertex clustering, the widely-used and well-known *stochastic block model* (SBM) [39] is of our particular interest. In a graph with stochastic block model, vertices are partitioned into several groups known as blocks, and the probability of connection between two vertices is solely determined by their block memberships. Despite its mathematical simplicity, SBM can well approximate any independent-edge random graphs with a sufficiently large number of blocks, and it is especially amenable to characterizing graphs with strong community structures. It has been shown that the clustering results obtained by certain methods will be almost accurate asymptotically if the random graph follows an SBM [27, 65, 66, 101]. We will discuss the details of stochastic block model in Chapter 3.

1.2 Spectral clustering methods

Among various vertex clustering approaches, we are most interested in the so-called spectral clustering methods. In general, *spectral clustering* refers to the techniques which make use of the spectral decomposition of some kind of *similarity matrix* that measures the similarities between data points. By spectral decomposition, the original data points are mapped through one-to-one correspondence to data points in another Euclidean space, on which the traditional Eucidean-based clustering methods, for example k-means, will be applied to finalize the clustering procedure. The new representation of the data points is usually in the form of rows of top (generalized) eigenvectors of the similarity matrix. By the properties of certain similarity matrix such as the Laplacian matrix, the clustering structure is enhanced so that traditional methods can more efficiently recover the clusters. As one of the most popular clustering methods, spectral clustering is easy to implement while it often outperforms traditional clustering algorithms under well-constructed similarity matrix. We refer the readers to [107] for a review of the spectral clustering.

In the context of clustering vertices of a graph there are two natural similarity matrices, namely the adjacency matrix and the Laplacian matrix of the graph. Based on these, numerous spectral clustering algorithms have been proposed to solve the vertex clustering problem [58, 87, 91, 101] under stochastic

block model. While the choice between adjacency matrix and Laplacian matrix is always debatable, it has been theoretically claimed that neither of them dominates the other in all cases [106]. In this thesis, we focus on the spectral method using adjacency matrix for ease of analysis. The reason is that some properties of the top eigenvectors of adjacency matrix, known as the *adjacency spectral embedding* (ASE), have been revealed in the literature [66, 101, 102], where it has been proven that the rows of ASE converge to some vectors, called *latent positions*, which fully determine the probability behavior of the stochastic block model. In this perspective, ASE can be regarded as a noisy version or estimate of the latent positions. In order to more clearly uncloak the relationship between the structure of adjacency spectral embedding and the parameters of stochastic block model, we consider the model of a *random dot product graph* (RDPG) [112], one type of well-studied latent position models. In RDPG, the probability of an edge between two vertices is just the inner product of the corresponding latent positions. As we will see in Chapter 3, a graph from stochastic block model can also be modeled as a random dot product graph with the number of distinct latent positions of RDPG equal to the number of blocks of SBM. Consequently, in this case spectral clustering can be interpreted as a method which clusters the vertices of a graph by detecting the group structure from the noisy version of their latent positions. Since the rows of ASE converge to their latent positions, the asymptotic accuracy of the clustering result

is theoretically ensured.

1.3 Model selection procedures

There are two inherent so-called *model selection* problems in spectral clustering. The first is determining the number of top eigenvectors whose rows are the low-dimensional points on which a traditional clustering method is applied. Since these top eigenvectors comprise the adjacency spectral embedding, we call such number the *embedding dimension*. The second is determining the number of clusters, which is usually required in algorithms such as k-means or in the method of Gaussian mixture model. Both of these problems need to be addressed before applying actual clustering procedure.

The first model selection problem of determining the embedding dimension has received a lot attention over the years. In more general scenarios we call the corresponding eigenvectors variables, thus the problem is called *variable selection*. The necessity of variable selection is based on the fact that only a part of the variables of the high-dimensional data are informative and important to the subsequent statistical inference. Using all the variables may not only lead to unnecessary computational cost, but may also decrease the performance of the clustering owing to the irrelevance or noise of some variables. Therefore, the selection of variables which optimize the clustering structure is

of great importance. Considering the overwhelming amount of methods on the topic of variable selection, we do not attempt to give a concise review of the literature. However, among various techniques of variable selection arguably the best-known methodology of principal component analysis (PCA) [47] is worth mentioning. In PCA, singular values of the data matrix are used to measure the importance of the variables, and the variables corresponding to relatively small singular values are discarded. For a broad review of the many stopping rules of PCA, we refer the readers to [43]. Unfortunately, there are no best rules in the task of dimension reduction in general due to bias-variance tradeoff. Roughly speaking, heuristic approaches are usually not theoretically reliable because they all need to determine a threshold which is highly subjective, while statistical approaches usually rely on an overly strong distributional assumption that the data does not often satisfy in many applications.

The second model selection problem, namely determining the number of clusters, is also a widely studied problem. As numerous approaches have been proposed on this topic, we refer the readers to the detailed reviews in [37, 75]. One substantial category of the methods is the *information criterion* approach. These methods evaluate and compare the so-called information criterion, usually some kind of penalized likelihood, on finite mixture models with different number of mixture complexity to perform model selection. Various information criteria are proposed, to list a few, such as Akaike information crite-

rion (AIC) [1], Bayesian information criterion (BIC) [94], an entropy criterion (NEC) [16], integrated completed likelihood (ICL) [8] and cross-validated likelihood [96]. Of these, we are mostly interested in BIC since it is a well-studied and easily implemented approach. Moreover, the consistency of estimation in number of components using BIC is theoretically supported in [49]. The practical performance of BIC approaches in model selection have also been highly rated by a large number of works [14,21,90,98]. In this thesis, we will consider the BIC approach as the solution to the traditional model selection problem in competition.

The traditional way to address both of the model selection problems in spectral clustering is to execute corresponding approaches successively. That is, applying spectral embedding with the dimension given by dimension reduction technique in the first step, then applying the model selection technique on the embedded data to estimate the number of clusters in the second step. This consecutive procedure of model selection suffers from three drawbacks. First, there are no best methods for estimating the embedding dimension. Even if we choose one of the modern and commonly used scree plot methods [115], in comparison the result is still not robust for limited data size (see detailed in Section 4.1.1). Second, the latter model selection procedure, namely estimating the number of clusters, completely depends on the result of the former one, because no information of the discarded variables will pass through. This may

cause an accumulation of errors when the former procedure performs poorly, even if the latter procedure is reliable. Third, the original data is truncated before applying clustering algorithm, which means it may not be possible to take advantage of any useful information contained in the discarded dimensions to improve the clustering result.

A breakthrough work of model selection in the framework of model-based clustering has been proposed in [88]. In this work, all of the variables are taken into consideration in a family of finite mixture models, which describes the distributional behavior of the raw data. The models are distinguished from each other by labeling all the variables as relevant, irrelevant or redundant, where different labeled variables follow distinct distributional structure in the model. The number of clusters, known as mixture complexity in mixture models, is also a factor which specifies a model within the family. The model selection procedure is conducted by comparing different models in the same family via Bayes factor, the ratio of the posterior probability of the model given the observations. A remarkable highlight of this framework is the simultaneity of selecting variables and number of clusters, which overcomes the drawbacks of the consecutive model selection procedure. This is the work upon which we make further improvements. We will provide more detailed discussion on the consecutive and simultaneous model selection procedure in Chapter 2.

1.4 Thesis contributions

In this thesis we develop models, theory as well as algorithms for vertex clustering on graphs generated from stochastic block model. Our work is inspired and established on the model-based clustering framework discussed above. Although the previous literature has built the groundwork of simultaneous model selection, it is not applicable to the vertex clustering task in the sense that neither the distributional model on the irrelevant variables nor the greedy variable selection algorithm is appropriate with respect to graph context. This calls for the development of a novel methodology of model selection and vertex clustering on the graphs with heterogeneous block structure. For this purpose, we will try to answer two questions throughout the thesis. First, what model should be used to properly characterize the distribution of spectral embedding when performing spectral clustering? Second, what model selection procedure should be conducted to ensure the accuracy of model estimation followed by subsequent clustering?

To answer the first question, the exact form of the spectral embedding needs to be clarified in a statistical perspective. We notice that an estimate of the embedding dimension is always required for the existing spectral embedding approaches. However, in practice the estimate may not be accurate for data with limited size, because the observed adjacency matrix is a noisy version of the

underlying edge probability matrix, whose spectral decomposition is the latent positions. The variety of the embedding dimension gives rise to uncertainty regarding the form of embedding. To avert the uncertainty, we propose a way of spectral embedding, called *extended adjacency spectral embedding* (extended ASE), in which the embedding is performed with a fixed dimension. The constant embedding dimension is chosen to be a loose upper bound of the dimension of latent positions. In the extended ASE, the variables are partitioned into an informative part, which corresponds to the variables with the true latent position dimension, and a redundant part, which is the set of remaining variables beyond true dimension. Under the framework of model-based clustering, we propose a family of specific Gaussian mixture models (GMM) to parameterize the entire extended ASE rather than only the informative part in some existing methods. The basis of the model comprises of two aspects. For the informative part, a state-of-the-art distributional result has been proven in [2], in which asymptotic mixed normality of the rows of ASE is stated. Meanwhile for the redundant part, an asymptotic mixed normality with the consistent mixture membership has been presented by strong evidence of our principled simulations. Merging these two perspectives leads to the first distributional model for the extended ASE. The details of the models will be discussed in Chapter 3.

To answer the second question, inspired by [88] we propose a simultaneous model selection (SMS) framework to address the issue occurring in the

consecutive model selection. The framework is specifically tailored for vertex clustering task on the graph with stochastic block model. It is conducted by comparing the integrated likelihood, the conditional probability of the observation given the model, of different models via BIC. In contrast with consecutive model selection procedure, our SMS identifies the embedding dimension, mixture complexity and membership of each vertex simultaneously without data cut-off. Moreover, we state and prove a theorem on the consistency of model parameter estimates. The theorem claims that the estimates in the model selection procedure given by our SMS method converge to the underlying truth for graphs with large size, provided the extended ASE follows the distribution in our proposed model. The theorem provides a theoretical support for the validity of our SMS method. Based on SMS, we also develop two heuristic algorithms to solve the vertex clustering problems, in which EM algorithm is employed in approximating the BIC. Finally, we evaluate the performance of the algorithms on a set of principled constructed data, which is generated to simulate two scenarios: 1) GMM, when data is assumed to follow our model; 2) SBM, when data is directly embedded from a graph. The superior performance of our algorithms in turn provides evidence of the efficacy of our GMM model. The details of SMS framework, theory, and algorithms will be discussed in Chapter 4.

Last but not least, we demonstrate our methodology on real data sets of con-

nectomes, a kind of graphs representing the neuronal connectivity in brains. We explain the variety of our algorithms in certain scenarios. For the noisy or corrupted data which may not fully follow the stochastic block model, we estimate model parameters through a regression technique which smooths the fluctuation of the BIC values. For directed graphs, we propose a new model to characterize the probabilistic behavior of the extended directed adjacency spectral embedding. The results adequately interpret the structural attributes of the connectomes. The details of the demonstration are presented in Chapter 5.

Chapter 2

Model-based clustering

In this chapter we provide a brief review of model-based clustering, including the modeling framework, the variable selection techniques and the method of selection of the number of components.

2.1 Notation

Throughout the thesis, $\mathbb{R}$ denotes the set of real numbers. $\mathbb{R}^{n\times D}$ denotes the set of matrices with dimension $n \times D$. $\mathbb{P}[\cdot]$ denotes the probability mass or probability density of the random variable/vector.

We use uppercase letters to denote data maxtrices, such as $X \in \mathbb{R}^{n\times D}$. $X_i \in \mathbb{R}^{1\times D}$ denotes the ith row of matrix X. We use letters with "hat", such as $\hat{X} \in \mathbb{R}^{n\times D}$ to denote the estimator of X. Lowercase bold letters denote random

vectors distributed by some probability distribution, such as $\boldsymbol{x} \sim p(\cdot; \Theta)$, and lowercase letters without boldface, such as $x_1, \ldots, x_D$, denote its corresponding entries.

2.2 Model-based clustering framework

In this section we provide a brief review of the framework of model-based clustering. For more details we refer the reader to [74]. The goal of *model-based clustering* [29] is to partition the set of data points into their own groups. Let $X \in \mathbb{R}^{n \times D}$ be the data matrix. Each row of X, namely X_i, is a realization, or say observation, of a D-dimensional random vector $\boldsymbol{x}$. In the setting of model-based clustering, we assume that $\boldsymbol{x}$ is distributed from a finite mixture probability distribution [26,71], whose components each represent a cluster that some data points from the same group belong to, i.e. the distribution of $\boldsymbol{x}$ can be written as

$$f(\boldsymbol{x}; \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi^{(k)} p(\boldsymbol{x}; \Theta^{(k)}) \tag{2.1}$$

Here, K represents the total number of components. $\boldsymbol{\pi} = (\pi^{(1)}, \ldots, \pi^{(K)})$ is the vector of prior probability, where $\pi^{(k)}$ is the mixing probability of the kth component, and satisfies $0 \leq \pi^{(k)} \leq 1$ for $k = 1, \ldots, K$. $\pi^{(k)}$ represents the prior probability or mixing proportion with which each observation x_i belongs to the kth component. Thus we also restrict $\sum_{k=1}^{K} \pi^{(k)} = 1$. $p(\cdot; \Theta^{(k)})$ is the probabil-

ity density function of the kth component, where $\Theta^{(k)}$ is the set of parameters corresponding to that component. $\boldsymbol{\Theta}$ denotes the set of all the parameters, i.e. $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \Theta^{(1)}, \ldots, \Theta^{(K)})$.

The first task of model-based clustering is model selection. Although there is no restriction that $p(\cdot; \Theta^{(k)})$ is a parametric function and $p(\cdot; \Theta^{(k)})$ are all from the same distribution family, we here assume all $p(\cdot; \Theta^{(k)})$ are from the same parametric model, for $k = 1, \ldots, K$. The selection of the distribution family of $p(\cdot; \Theta^{(k)})$ will be discussed in chapter 3. Now we assume that the form of $p(\cdot; \Theta^{(k)})$ has been chosen already, for example, a multivariate Gaussian distribution. Then there are two problems of model selection that concern us. One is the variable selection, in which we pick the variables or determine the structure of $\Theta^{(k)}$ that possesses clustering information; the other is the selection of the number of components K. We will discuss the model selection procedure in detail later.

Once the model has been selected, the parameters of the model are usually fitted through *Maximum Likelihood Estimation* (MLE), i.e.

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\Theta}; X) \tag{2.2}$$

where $X \in \mathbb{R}^{n \times D}$ is the data matrix whose rows are a realization of the D dimensional random vector $\boldsymbol{x}$, and $\mathcal{L}(\boldsymbol{\Theta}; X)$ is the likelihood function of n data

points presented in X. If the n data points are independently identically distributed (i.i.d), then the likelihood function will be

$$\mathcal{L}(\boldsymbol{\Theta}; X) = \prod_{i=1}^{n} f(X_i; \boldsymbol{\Theta}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi^{(k)} p(X_i; \Theta^{(k)}) \tag{2.3}$$

Considering the complicated and multi-modal form of the joint likelihood, the analytical closed-form solutions to (2.2) are usually impractical. Therefore a commonly used procedure, the *Expectation Maximization* (EM) algorithm [3, 23, 70], is used to find the MLE. More discussion on EM algorithm will be presented in chapter 4.

Let $\hat{\boldsymbol{\Theta}} = (\hat{\boldsymbol{\pi}}, \hat{\Theta}^{(1)}, \ldots, \hat{\Theta}^{(K)})$ be the estimation of the parameters of the model. Then we can calculate *a posteriori* probability $\hat{\boldsymbol{z}}_i = (\hat{z}_{i1}, \ldots, \hat{z}_{iK})$ for data point X_i by

$$\hat{z}_{ij} = \frac{\hat{\pi}_j p(X_i; \hat{\Theta}_j)}{\sum_{k=1}^{K} \hat{\pi}_k p(X_i; \hat{\Theta}_k)} \tag{2.4}$$

for $j = 1, \ldots, K$. The a posteriori probability $\hat{z}_{ij}$ can be explained as the probability that the ith data point belongs to the jth component after seeing the observation. Notice that $\sum_{j=1}^{K} \hat{z}_{ij} = 1$. For the purpose of clustering, each data point is usually assigned to a corresponding cluster by the *maximum a posteriori* (MAP) rule [72], i.e. the ith data point is assigned to the kth component if

$$k = \arg\max_{j} \{\hat{z}_{ij}\} \tag{2.5}$$

We do this for $i = 1, \ldots, n$ so that the whole n data points are clustered by the model-based clustering.

2.3 Selection of number of components

Determining the number of groups in a data set is a widely studied problem over the years, since many well-known clustering methods, for example, *k-means* [63], require the number of clusters to be specified. Numerous approaches have been proposed on this topic. We refer the readers to the detailed reviews provided in [37,75]. Some other examples of recent works include minimum description length (MDL) based method [9], X-means [81], rate distortion theory based approach [100] and G-means [36].

One substantial category of methods, the information criteria methods, assume the data from a specific parametric model, on which some kind of penalized likelihood can be evaluated and compared. Various information criteria are proposed, such as Akaike information criterion (AIC) [1], Bayesian information criterion (BIC) [94], an entropy criterion (NEC) [16], integrated completed likelihood (ICL) [8] and cross-validated likelihood [96]. Of the foregoing, the BIC approach is well-studied and easily implemented. The consistency of estimation in number of components using BIC has been shown in [49]. A large number of works have also shown that selecting the model by comparing the

BIC yields good performance in many applications [14,21,90,98]. Moreover, as will be discussed below, BIC can be used to approximate the integrated likelihood in the model selection procedure of interest [29]. This is in accordance with the theoretical analysis of the model selection framework. Therefore, in this thesis we will focus on BIC approaches while considering model selection. We now present the framework of BIC-related model selection in model-based clustering.

The idea of selection of number of components within model-based clustering, one specific task of model selection, is summarized in [29]. The model selection procedure is based on comparing the posterior probability of different models. Mathematically, let the two models that we are going to compare be M_1 and M_2. In the context of selection of number of components, M_1 and M_2 could be two finite mixture models with different numbers of components. But more generally, the difference between M_1 and M_2 is not limited to the number of components. We will discuss the variable selection problem within the same framework in the next section. According to Bayes' theorem, the posterior probability of each model given the observation data X will be proportional to the product of the prior of the model and conditional probability of X under the model, i.e.

$$P(M_i|X) \propto P(M_i)P(X|M_i) \tag{2.6}$$

for $i = 1, 2$. In formula (2.6), $P(M_i)$ is the prior probability of model M_i. By

the law of total probability, the quantity $P(X|M_i)$, known as the integrated likelihood, can be obtained by integrating over all the unknown parameters in the model, i.e.

$$P(X|M_i) = \int P(X|\theta_i, M_i)P(\theta_i|M_i)d\theta_i \tag{2.7}$$

where θ_i is the set of parameters in model M_i. If we lack the knowledge of the prior probability of the models, which is often the case in practice, we would assume all priors are the same. Then comparing the posterior probability of the models given the data is equivalent to comparing the corresponding integrated likelihoods. The *Bayes factor*, which is defined as the ratio of two integrated likelihoods

$$B_{12} = P(X|M_1)/P(X|M_2) \tag{2.8}$$

is used to determine which model is in favor. In practice, however, the calculation of integrated likelihood is usually very difficult. So alternatively, the quantity $P(X|\theta_i, M_i)$ is approximated by the *Bayesian information criterion* (BIC) [94], that is

$$2\log P(X|M_i) \approx 2\log P(X|\hat{\theta}_i, M_i) - \eta_i \log n \tag{2.9}$$

where $\hat{\theta}_i$ is the MLE of the parameters under M_i, η_i is the number of parameters, and n is the number of observations.

2.4 Variable selection

The problem of variable selection has received a lot of attention in many fields. It is often the case that only a few dimensions of the high-dimensional data are informative and important to the statistical inference and analysis. Using all the dimensions (in some scenarios we called them variables or features) may not only render more unnecessary computational cost, but decrease the performance of the learning process from the irrelevant, redundant or noisy dimension as well. To overcome this, the variable selection procedure is of great necessity and importance.

2.4.1 Dimension reduction via measure of importance

If along with the data itself we have a measure of importance associated with each dimension/variable, then we can use these measures to determine which variables need to be retained without looking at the group structure of the data. This is usually a good approach if most clustering information is contained in the variables with larger measures of importance. In this case, since we simply discard the variables with smaller measures of importance, we also call such a variable selection procedure *dimension reduction*.

Among the various techniques of dimension reduction, principal component

analysis (PCA) [47] is probably the best-known one. In PCA, the measure of importance is just the singular values of the data matrix. A broad review of the many stopping rules in PCA is summarized in [43]. These methods can be roughly categorized into heuristic approaches or statistical approaches. Although many methods perform well in specific cases, there are obvious drawbacks to each of them. Roughly speaking, heuristic approaches, for example, Kaiser-Guttman criterion [34], Broken-stick [30], proportion of total variance [47] and scree plots [114], are usually not theoretically reliable because they all need to determine a threshold which is highly subjective. On the other hand, statistical approaches, for example, Bartlett's test of equality [4], Bartlett's test of sphericity [82], Lawley's test [55] and bootstrap methods, usually rely on an overly strong assumption that the data is independently and identically distributed as some specific distribution, which is often not satisfied in many cases.

Among all the alternatives, choosing the dimension by *scree plot* is an easily implemented and ubiquitous approach by singular value thresholding (SVT). The key is to plot the eigenvalues or singular values from spectral decomposition in descending order, and then locate the "elbows" which divide the eigen/singular values into a signal part and a noise part. This method works well if the eigen/singular values of the signal dimensions are well separated in magnitude from those of the noise dimensions. But for noisy or corrupted data,

the "gap" is often not obvious. Although there are various criteria about this, deciding the "elbow" in the scree plot is still a highly subjective procedure. In general, there are no "best" methods in the area of dimension reduction. Some methods are well justified by theoretical proofs, for example the universal SVT method [18], but they may not perform well in practice because of finite data. In section 4.1.1, we will discuss a simple but effective method, which has been proposed in [115], to determine where the elbow is in the scree plot. This method shows relatively good performance among the existing approaches, thus it is used as a comparison for our new methods.

2.4.2 Variable selection via group structure

It has been shown that, in general, principal components with larger eigenvalues from PCA do not necessarily contain more clustering information [17]. Because of this, other than the measure of importance for each dimension, researchers seek methods of variable selection by considering their group structure. That is, selecting the relevant variables/features that are informative to the learning process. Various aspects of the feature selection problem has been studied for a long time. For an extended account of these studies, we refer the readers to [11, 46, 51–53, 60–62, 113]. In this thesis, we are interested in the variable selection problem with model-based clustering.

In model-based clusering, the variable selection problem is restated as de-

termining the relevance of the variables in the group structure [89]. In this manner, a variable can be classified as *relevant*, *redundant* or *irrelevant*. A relevant variable contains information about the group structure, thus its realization highly depends on the membership of the corresponding data point. A redundant variable may contain similar information, but because it is dependant on the relevant variables it will not provide additional information for clustering given the relevant variables. An irrelevant variable, on the other hand, does not contain any useful information for clustering. Its distribution is independent from the membership of the data point thus it could be regarded as noise in the clustering tasks.

A comprehensive review of the variable selection approaches in the model-based clustering framework has been provided in [28]. The author categorizes the methods into Bayesian approaches, penalization approaches or model selection approaches. In the class of Bayesian approaches, a latent variable, which characterizes the state of relevance of its corresponding variable, is introduced. Those methods determine the relevance of the variables by calculating the posterior distribution of the latent variables. In the class of penalization approaches, a penalized log-likelihood is used for each variable. The penalty term is constructed in order to indicate if a variable has any significant contribution to the clustering tasks. Tremendous efforts have been dedicated to these two classes of methods, leading to vast literature on the studies of Bayesian ap-

proaches [6, 54, 104] and penalization approaches [12, 32, 79, 108, 111].

Unlike the Bayesian approaches and penalization approaches, the model selection approaches take the relevant, redundant and irrelevant variables into account in the probabilistic model. That is, redundant and irrelevant variables also play roles in the model, but their roles are different from that of the relevant variables. Model selection approaches will adjust the model for all the variables according to the variable selection result. This could explain why those methods are classified in the category named "model selection".

Following the work of [29], a breakthrough in the area of variable selection for clustering is the work of [88], which has proposed a variable selection framework by model selection and a corresponding algorithm. Let $x \in \mathbb{R}^D$ be the vector containing all the variables of interest, and $X \in \mathbb{R}^{n \times D}$ be the observed data matrix with n observations. The author partitioned all the variables into three sets, x^S, x^N and x^C. x^S denotes the variables which have already been selected as they may contain clustering information. x^N denotes the variables which have not been selected as they may be irrelevant or redundant. x^C denotes the variables which are the candidates for being selected. Then the model selection procedure states two models M_1 and M_2 to be compared. M_1 is the model that x^C is independent with the cluster memberships conditioned on x^S, while M_2 is the model that x^C does depend on cluster memberships conditioned on x^S. Thus in M_1, because x^C does not provide any additional clustering information,

it will be discarded; while in M_2, because x^C helps the clustering, it will be selected. The model selection will be conducted by comparing the integrated likelihood of the two cases. Mathematically, let X^S, X^N and X^C be the observations on the corresponding variables. It follows that the integrated likelihood would be

$$P(X|M_1) = P(X^N|X^S, X^C, M_1)P(X^C|X^S, M_1)P(X^S|M_1) \tag{2.10}$$

$$P(X|M_2) = P(X^N|X^S, X^C, M_2)P(X^C, X^S|M_2) \tag{2.11}$$

Under the assumption that $P(X^N|X^S, X^C, M_1) = P(X^N|X^S, X^C, M_2)$, the Bayes factor defined in (2.8) would be

$$B_{12} = \frac{P(X^C|X^S, M_1)P(X^S|M_1)}{P(X^C, X^S|M_2)} \tag{2.12}$$

A specific model needs to be constructed for both M_1 and M_2 in order to perform model-based clustering. For example, the author assumes $X^S|M_1$ and $X^C, X^S|M_2$ have a Gaussian mixture distribution, while $X^C|X^S, M_1$ just corresponds to a linear regression. This could be explained as the result that X^C does not provide additional clustering information under M_1. Considering the difficulty of calculating the integrated likelihoods, as usual we can approximate

their logarithm by BIC, as mentioned in (2.9). That is, we can compare

$$\text{BIC}_1 = \text{BIC}(X^C|X^S, M_1) + \text{BIC}(X^S|M_1) \tag{2.13}$$

$$\text{BIC}_2 = \text{BIC}(X^C, X^S|M_2) \tag{2.14}$$

to evaluate the two models. In this manner, any set of variables X^C could be tested to see if it has significant clustering information, from which we can decide if it needs to be selected as the clustering variables. Obviously, the number of choices of X^C is of combinatorial complexity. Thus the author has proposed a greedy search algorithm [88], by which a candidate variable could be added to or removed from the set of selected variables through the comparison of BIC. Notice that either the framework or the algorithm above can be combined with the model selection procedure discussed in section 2.3 to select the number of components at the same time.

Based on the above work, further effects on model selection approaches have been made, including the extension of the framework [31, 68, 69] and improvement of the algorithm [67, 95]. The evaluation of the performance of the above methods can be found in [15, 99]. It has been shown that model selection approaches for variables selection have advantage over other approaches in many aspects. In summary, the model selection approaches provide us a way to characterize all the variables whose underlying distribution is embedded in the

statistical model itself. A remarkable highlight is that they select variables and number of clusters simultaneously rather than successively, as in traditional methods. Intuitively, clustering can benefit from the simultaneity in the sense that, while both factors affect each other, global optimal wins over local optimal.

Chapter 3

Models for Extended Adjacency Spectral Embedding

In this thesis, the problem of interest is the clustering of graph vertices, provided that the vertices from the same group share some common connection behavior, which is distinguishable from that of the vertices in other groups. Features of the graph need to be extracted before clustering tasks are conducted. There are some existing unsupervised methods of feature extraction from graphs, for example, by computing summary topological and label statistics [59, 80], by frequent subgraph mining algorithms [40, 45], or by treating each edge of a graph as a raw feature using PCA. However, these methods ignore the topological structure of the graphs and thus suffer from the lack of knowledge of intrinsic clustering patterns. In order to overcome this shortcom-

ing, effectively modeling the real-world networks of interest is of great appeal and importance.

In this chapter, we provide a brief review of the random graph models which capture the heterogeneous vertex attributes in vertex clustering problems. The corresponding spectral embedding is presented as the estimators or test statistics for the subsequent inference. With state-of-the-art consistency and distribution results, in addition to the result of simulation, we provide a first probability model for the spectral embedding of random graphs. This new model is what our model-based clustering task is based on.

3.1 Random graphs

We first present the notation of the graph and its corresponding adjacency matrix that we use throughout this thesis:

> **Definition 1** (Unweighted and weighted graph)**.** We use $G = (V, E)$ to denote an *unweighted graph*, where V is the set of *vertices* and E is the set of *edges*. If G has n vertices, we usually represent V as $[n] = \{1, 2, \ldots, n\}$, and we represent E as a subset of $[n] \times [n]$. There is an element $(i, j) \in E$ if and only if there is an edge from vertex i to vertex j. If for each edge e in the graph, there is a *weight* w_e assigned to it, we call such graph a *weighted graph* denoted by $G = (V, E, \{w_e\}_{e \in E})$.

Definition 2 (Adjacency matrix)**.** For an unweighted graph $G = (V, E)$ with $V = [n]$, its *adjacency matrix* $A = [A_{ij}] \in \{0, 1\}^{n \times n}$ is defined by

$$A_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E \\ 0, & \text{if } (i, j) \notin E \end{cases} \tag{3.1}$$

For a weighted graph $G = (V, E, \{w_e\}_{e \in E})$, its *adjacency matrix* $A \in \mathbb{R}^{n \times n}$ is defined by

$$A_{ij} = \begin{cases} w_{(i,j)}, & \text{if } (i, j) \in E \\ 0, & \text{if } (i, j) \notin E \end{cases} \tag{3.2}$$

It is easy to see that a graph G can be fully characterized by its adjacency matrix A, so we will not distinguish between a random graph and its adjacency matrix.

3.1.1 Inhomogeneous Erdős-Rényi graph

In this thesis our focus is the statistical inference on unweighted random graphs. Due to both simplicity and previous theoretical work, we always assume that the edges in the graph exist independently of each other. That is, whether there is a edge between vertex i and j does not depend on the connectivity of other vertices. One of the simplest generative models for random

graphs is the *Erdős-Rényi* (ER) graph [25], where the edges are independent with a common probability. To deal with more general cases, we consider the *inhomogeneous Erdős-Rényi* (IER) graph, an extension of the original ER graph, where the edges are still independent but may arise with their own probability. Those probabilities are entries of an $n \times n$ *edge probability matrix* P. The definition of IER is as follows:

Definition 3 (Inhomogeneous Erdős-Rényi (IER) model)**.** G is an *inhomogeneous Erdős-Rényi* (IER) graph with n vertices, denoted by $G \sim \text{IER}(P)$, if its edges are generated according to an *edge probability maxtrix* $P = [P_{ij}] \in [0,1]^{n\times n}$. That is, A_{ij}, the entry of the adjacency matrix A, follows an independent Bernoulli distribution with parameter P_{ij}, i.e.

$$\mathbb{P}[A_{ij}] = P_{ij}^{A_{ij}}(1 - P_{ij})^{1-A_{ij}} \tag{3.3}$$

for all $(i, j) \in [n] \times [n]$.

3.1.2 Random dot product graph

The introduction of the edge probability matrix P in this definition allows us to describe any models for unweighted random graphs whose edges are independent of others. To more effectively depict the heterogeneous attributes

of the vertices in the graph, we consider a map from the set of vertices onto a latent space. In the so-called *latent position graph* [38], the probability of the connection between vertex i and j, namely the i, j-th entry of the edge probability matrix P, depends only on the two latent positions of vertex i and j. In other words, the latent positions contain comprehensive information about every vertex that gives rise to the probabilistic connection between them. Among the various latent position models, we are particularly interested in the *random dot product graph* (RDPG) [112], a well-studied and practical latent position graph. In RDPG, the latent space is a subspace of Euclidean space with dimension d, and the latent positions, which we call latent vectors, are d-dimensional vectors in that subspace. The probability of an edge is simply the inner product of the corresponding latent vectors, and this is where the name RDPG comes from. For an undirected graph, we can represent all the latent positions by an $n \times d$ matrix X, where the ith row of X is the latent vector of the ith vertex. Then by the definition of inner product, the edge probability matrix is simply given by $P = XX^T$. The definition of undirected RDPG is as follows:

> **Definition 4** (Random dot product graph (RDPG))**.** Let $\mathcal{X} \subset \mathbb{R}^d$ be a subset of $\mathbb{R}^d$ satisfying $x^T y \in [0, 1]$ for all $x, y \in \mathcal{X}$. Let $X_1, \ldots, X_n \in \mathcal{X}$ be n latent vectors, and $X \in \mathbb{R}^{n \times d}$ be the *latent position matrix* such that the ith row of X is X_i. If the edges of an undirected graph G are generated according to an edge probability matrix $P = XX^T$, then we say G is a *random dot product*

graph (RDPG) with latent position matrix X, denoted by $G \sim \text{RDPG}(X)$. That is, A_{ij}, the entry of the adjacency matrix A, follows an independent Bernoulli distribution with parameter $P_{ij} = X_i^T X_j$, i.e.

$$\mathbb{P}[A_{ij}] = (X_i^T X_j)^{A_{ij}}(1 - X_i^T X_j)^{1-A_{ij}} \tag{3.4}$$

for all $(i, j) \in [n] \times [n]$.

For directed graphs, we will also have a corresponding version of RDPG. The modification is that each vertex has two latent positions, one for in-neighbor, say $(X_{\text{in}})_i \in \mathbb{R}^d$, and one for out-neighbor, say $(X_{\text{out}})_i \in \mathbb{R}^d$. Thus $X_{\text{in}} \in \mathbb{R}^{n\times d}$ and $X_{\text{out}} \in \mathbb{R}^{n\times d}$ will be two latent position matrices, and the edge probability matrix $P = X_{\text{in}} X_{\text{out}}^T$.

3.1.3 Stochastic block model

In the context of vertex clustering, vertices from the same group are supposed to share common connection attributes. Therefore, in latent position models we may assume vertices from the same group have the same latent position. This leads to the *stochastic block model* [39], in which the set of vertices is partitioned into K groups called blocks. The connection of the graph is parameterized by the *block connectivity probability matrix* B, which solely

determines the edge probability within and between blocks. The ER graph is a simple example of SBM with one single block. The formal definition of SBM is given below:

Definition 5 (Stochastic block model (SBM))**.** Let G be the graph of interest with n vertices, $B \in [0,1]^{K\times K}$ be the block connectivity probability matrix, and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K) \in (0,1)^K$ be the vector of prior block probability such that $\sum_{i=1}^{K} \pi_i = 1$. G is called a K-block *stochastic block model* (SBM) graph, denoted by SBM$(n, B, \boldsymbol{\pi})$, if there is a random vector $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)$, called the *block memberships*, that assigns vertex i to block k with probability π_k. Mathematically, $\tau_1, \ldots, \tau_n$ are i.i.d. random variables with categorical distribution and with parameter $\boldsymbol{\pi}$, i.e.

$$\mathbb{P}[\tau_i = k] = \pi_k \tag{3.5}$$

for all $i \in [n]$ and $k \in [K]$. Furthermore, the edges are generated according to an edge probability matrix P, whose i,j-th entry is B_{τ_i,τ_j}. Equivalently, A_{ij}, the entry of the adjacency matrix A, follows an independent Bernoulli distribution with parameter $P_{ij} = B_{\tau_i,\tau_j}$, i.e.

$$\mathbb{P}[A_{ij}] = (B_{\tau_i,\tau_j})^{A_{ij}}(1 - B_{\tau_i,\tau_j})^{1-A_{ij}} \tag{3.6}$$

for all $(i, j) \in [n] \times [n]$.

In addition, it is convenient to consider that the block memberships vector τ is not random but rather fixed in some cases. We call such a graph an *SBM conditioned on block memberships*, denoted by SBM$(B, \boldsymbol{\tau})$. If an undirected graph $G \sim$ SBM$(B, \boldsymbol{\tau})$ and B is positive semidefinite, then G can be represented by a RDPG with at most K distinct latent positions. To see this, let $B = U_B \Lambda_B U_B^T$ be the eigen-decomposition, where $\Lambda \in \mathbb{R}^{d \times d}$ is the diagonal matrix of nonzero eigen-values and $U_B \in \mathbb{R}^{K \times d}$ is the matrix of corresponding eigen-vectors. We can map each vertex to a latent vector $(U_B \Lambda_B^{\frac{1}{2}})_k$, the kth row of $U_B \Lambda_B^{\frac{1}{2}}$, if the vertex is assigned to the kth block by $\boldsymbol{\tau}$. In this case, all vertices in the same block have the same latent vectors. Similarly, for a directed graph of SBM, we can perform singular value decomposition (SVD) $B = U_B \Sigma_B V_B^T$ to get the in-latent position matrix $X_{\text{in}} = U_B \Sigma_B^{\frac{1}{2}}$ and out-latent position matrix $X_{\text{out}} = V_B \Sigma_B^{\frac{1}{2}}$. In this formula, $\Sigma_B \in \mathbb{R}^{d \times d}$ is the diagonal matrix of nonzero singular values, $U_B \in \mathbb{R}^{K \times d}$ is the matrix of corresponding left singular vectors, and $V_B \in \mathbb{R}^{K \times d}$ is the matrix of corresponding right singular vectors. This builds a connection between the SBM with a positive semidefinite block connectivity probability matrix and the RDPG. For the relationship between an SBM with a non-positive semi-definite block connectivity probability matrix and a generalized RDPG, we refer the readers to [93].

The relationship between IER, RDPG, SBM and ER is shown in figure 3.1. In this thesis we only consider the random graph models in which edges arise independently, so all the models we have discussed above are IER. Among the various latent position models, RDPG is a simple, tractable and well-studied option in which many existing techniques including classical statistics and linear algebra are useful to graph inference. To capture community structures, SBM is frequently used to approximate many real-world graphs. Therefore, we are mostly interested in cases in which an SBM graph with positive semidefinite block connectivity probability matrix also follows a RDPG model. Finally, ER is a very special SBM with just one blcok.

3.2 Spectral embedding

Now we consider graphs that follow the SBM in accordance with our clustering problems. Given an observed SBM graph, our inference task is to identify the underlying memberships of the vertices corresponding to the blocks that they belong to. That is, if $G \sim \text{SBM}(B, \boldsymbol{\tau})$, our goal is to infer the graph parameter τ from the observed adjacency matrix A. Inspired by the paradigm in classical statistical inference tasks, many statistical inference procedures for graphs seek a representation of the vertices in the Euclidean space from the observation of the graph so that the data contains the information for the in-

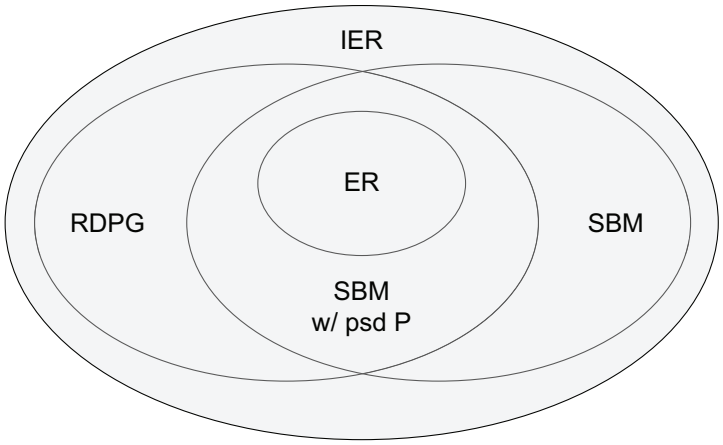


Figure 3.1: The relationship between random graph models IER, RDPG, SBM and ER. The models that we are interested in are all IER models, in which the edges are independent according to an edge probability matrix. RDPG is a tractable latent position model, where classical statistics and linear algebra techniques can be useful to analyze the graph inference. Vertices are partitioned into blocks in SBM, which is frequently used to capture community structures for many real-world problems. If an SBM graph has a positive semidefinite block connectivity probability matrix, then it also can be modeled as a RDPG. Specifically, ER is simply a 1-block SBM.

ference task. Among the various techniques, spectral methods are effective, well-studied and computationally feasible approaches that convert the data of a graph into vectors in Euclidean space. So in this thesis, we will focus on spectral methods, especially *Adjacency spectral embedding* (ASE), for the inference on RDPG models.

3.2.1 Adjacency spectral embedding

Spectral methods are the methods that depend on spectral decomposition of some matrix that represents the graph. The word embedding is used to denote the matrix obtained from the spectral decomposition, whose rows represent the corresponding vertices in the graph. If the matrix on which spectral decomposition is applied is just the adjacency matrix of the graph, we call the resulting matrix *adjacency spectral embedding* (ASE) [101]. The formal definition of ASE is as follows:

Definition 6 (Adjacency spectral embedding (ASE))**.** Let G be an undirected graph of interest with n vertices, and $A \in \mathbb{R}^{n \times n}$ be its symmetric adjacency matrix. Let the spectral decomposition of A be

$$A = \hat{U}\hat{\Lambda}\hat{U}^T \tag{3.7}$$

Here, $\hat{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with eigenvalues of A on its diagonal in descending order. That is, $\hat{\Lambda} = \text{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_n)$ with $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_n$. $\hat{U}$ is an orthogonal matrix whose columns are corresponding eigenvectors of A. For a given integer d satisfying $1 \leq d \leq n$, called *embedding dimension*, the *adjacency spectral embedding* (ASE) of G with dimension d is given by

$$\hat{X} = \hat{U}_{[d]}\hat{\Lambda}_{[d]}^{\frac{1}{2}} \tag{3.8}$$

> where $\hat{U}_{[d]} \in \mathbb{R}^{n\times d}$ is the submatrix of $\hat{U}$ with its first d columns, and $\hat{\Lambda}_{[d]} \in \mathbb{R}^{d\times d}$ is the submatrix of $\hat{\Lambda}$ with its first d rows and columns.

In this definition, the ASE $\hat{X} \in \mathbb{R}^{n\times d}$ is the matrix containing top d eigenvectors normalized by the square root of corresponding eigenvalues. Notice that $\hat{X}\hat{X}^T = A$. The motivation of ASE is to provide an estimation of the latent position matrix if the graph follows RDPG. Specifically, let G be a RDPG with latent position matrix $X \in \mathbb{R}^{n\times d}$. If G is also an SBM graph with K blocks, as we demand in clustering tasks, then X is a matrix with each row being one of the K distinct latent positions. Obviously X has all the information for clustering the vertices into corresponding blocks, therefore estimating X by the observation A is extremely helpful. By definition 4, the edge probability matrix $P = XX^T \in \mathbb{R}^{n\times n}$. Let $P = U\Lambda U^T$ be the spectral decomposition of P, where $\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix with eigenvalues of P on its diagonal in descending order, and U is an orthogonal matrix whose columns are corresponding eigenvectors of P. If $\text{rank}(P) = d$, then only d eigenvalues in Λ is nonzero. Thus the latent position matrix can be written as

$$X = U_{[d]}\Lambda_{[d]}^{\frac{1}{2}} \tag{3.9}$$

By definition 3, A can be regarded as a realization of P, with $\mathbb{E}[A] = P$. In fact, it has been shown that A is asymptotically close to P with high probability [58,

64,78]. Thus it is natural to believe that the spectral decomposition of A is close to the spectral decomposition of P with the same dimension. This intuition is theoretically supported by [48] for the closeness between eigenvalues of A and P, and by [22] for the closeness between eigenspaces of A and P. We will discuss more recent work on the consistency result of ASE later.

3.2.2 Embedding dimension

The choice of the embedding dimension d in definition 6 is one of the main problems that we are interested in throughout this thesis. Without loss of generality, we may assume the latent position matrix X is of full column rank in RDPG. Otherwise we can always reformulate it, for example by spectral decomposition of P, without changing the structure of the distribution, or say P. Thus the rank of $P = XX^T$ will equal the number of columns of X, i.e. $\text{rank}(P) = \text{rank}(X) = d$. Ideally, the embedding dimension should be chosen as the dimension of the latent position d. In real data applications, however, d is unknown because P is unobserved and the only observation A is a noisy version of P. Practically, we apply the embedding up to dimension D, where D is an integer which is believed to be an estimation of the upper bound of d under the specific application. Based on the embedding result with D dimension, as we will discuss later in chapter 4, various variable selection approaches can be applied to estimate the true d. If $\hat{d}$ is the estimator of d, one can either truncate

the embedding matrix up to $\hat{d}$ dimension or treat $\hat{d}$ as a model parameter in the subsequent inference task. So we assume the upper bound D of the dimension of latent position is always given, and D is the dimension by which we perform ASE in the first place.

Another ambiguity in definition 6 may arise when computing $\hat{\Lambda}_{[d]}^{\frac{1}{2}}$ in (3.8). As we discussed previously, the latent position matrix $X = U\Lambda^{\frac{1}{2}}$, where $P = U\Lambda U^T$ is the spectral decomposition. If G is RDPG, then all the eigenvales of P are nonnegative since P is positive semi-definite. Thus $\Lambda^{\frac{1}{2}}$ is the matrix by taking square root of Λ entry-wise, i.e. $\Lambda^{\frac{1}{2}} = \text{diag}(\lambda_1^{\frac{1}{2}}, \ldots, \lambda_n^{\frac{1}{2}})$. However, there is no guarantee that A, the perturbed version of P, is also positive seme-definite. So it is possible that some eigenvalues of A are negative. To fix the ambiguity, we let $\hat{\Lambda}$ be the diagonal matrix with the absolute values of the eigenvalues of A on its diagonal in descending order. That is, we sort the eigenvalues by magnitude. For large-scale data, this modification is almost equivalent to the original one in the sense that the top d eigenvalues of A are nonnegative and the remaining ones are close to zero with high probability for sufficiently large n, by Weyl's inequality and closeness between A and P [64].

3.2.3 Comparison with Laplacian spectral embedding

There are numerous other matrices on which the spectral decomposition is performed for graph embedding [57]. We hereby mention the *Laplacian spectral embedding* (LSE). LSE is based on the spectral decomposition on the *normalized Laplacian matrix* [19], which is defined as

$$\mathcal{L} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{3.10}$$

where $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix of the graph, and $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal entries are $D_{ii} = \sum_{j=1}^{n} A_{ij}$. Notice that this definition is different from the definition of *combinatorial Laplacian matrix*, which is used in the well-known Laplacian eigenmaps [5]. The definition of LSE is the same as that of ASE, except for replacing the matrix of A with $\mathcal{L}$ in the spectral decomposition step. The comparison of ASE and LSE in the subsequent inference task of recovery block assignments for a graph of SBM is discussed in [106]. The author has constructed a measure by Chernoff information to evaluate the relatively large-sample performance of ASE compared to LSE. By this measure, it has been theoretically claimed that neither ASE or LSE dominates the other over the whole parameter space in the inference task of clustering. In general, it has been observed that LSE dominates ASE

when the block connectivity probability matrix B is sparse, and ASE dominates LSE when the entires of B are large. So in this thesis, we will focus on the models based on ASE, under the consideration that ASE is both intuitively and theoretically an approximation of the latent position matrix on which the subsequent inference task relies.

3.3 Asymptotical properties of extended adjacency spectral embedding

Considering inference tasks of vertex clustering in stochastic block model, we notice that the latent position matrix X has perfect block membership information. While X can be viewed as spectral embedding of the edge probability matrix P by $P = XX^T$, X is unknown since P cannot be observed in practice. On the other hand, as we discussed previously, the adjacency matrix A can be regarded as a noisy version of P in any inhomogeneous Erdős-Rényi graph. It has been shown that the spectral norm of their difference, namely $||A - P||_{2\to 2}$, is well controlled by several upper bounds [58, 64, 78]. From the closeness between A and P, it is intuitive to believe that $\hat{X} = \hat{U}_{[d]}\hat{\Lambda}_{[d]}^{\frac{1}{2}}$ by (3.8) is close to $X = U_{[d]}\Lambda_{[d]}^{\frac{1}{2}}$ by (3.9). In fact, the upper bounds for both the Frobenius norm $||\hat{X} - X||_F$ [101, 102] and $2 \to \infty$ norm $||\hat{X} - X||_{2\to\infty}$ [66] have been proven. So it is natural to consider using the estimation of X, namely the adjacency spectral

embedding $\hat{X}$, in the inference task.

Many efforts have been made on the consistency results of adjacency spectral embedding for estimating the block memberships. It has been shown in [101] that the number of misassigned vertices by clustering the rows of ASE using k-means algorithm is $O(\log n)$. Another consistency result states that the number of misassigned vertices in clustering by using ASE is almost always less than n^{ϵ}, for any $\epsilon > \frac{3}{4}$, even if the true rank of P is unknown [27]. Moreover, it has been proven that almost perfect clustering is possible by clustering the rows of ASE for SBM under some eigengap assumptions [65]. The upper bounds for the difference between the embedding $\hat{X}$ and the true latent positions X up to rotation are also provided, both in Frobenius norm [105] and $2 \to \infty$-norm [66].

While the previous consistency results demonstrate the presence of the underlying block membership structure in adjacency spectral embedding, they cannot be applied to model-based clustering in which the distributional knowledge is necessary. But is it enough to know the distribution of ASE to apply model-based clustering? The answer depends on whether we know the true embedding dimension d. As mentioned in section 3.2.2, d is unknown in real-world applications. As a result, adjacency spectral embedding is usually performed on a dimension larger than the true one. That is why we need extra distributional knowledge rather than the knowledge of standard ASE. To be specific, let

$D(\geq d)$ be an estimated upper bound of d. Usually this upper bound is inferred from the prior information of the application, and it is independent with each individual edge probability matrix P of the graph. We then apply adjacency spectral embedding on the adjacency matrix A according to definition 6, except that the embedding dimension is set to be D instead of d. We call the resulting matrix with dimension D *extended adjacency spectral embedding* (extended ASE), denoted by

$$\hat{Z} = \hat{U}_{[D]}\hat{\Lambda}_{[D]}^{\frac{1}{2}} \tag{3.11}$$

where $\hat{U}_{[D]} \in \mathbb{R}^{n\times D}$ is the submatrix of the eigenvector matrix $\hat{U}$ with its first D columns, and $\hat{\Lambda}_{[D]} \in \mathbb{R}^{D\times D}$ is the submatrix of diagonal eigenvalue matrix $\hat{\Lambda}$ with its first D rows and columns. From the formula, it is trivial that the first d columns of $\hat{Z}$ is the regular ASE $\hat{X}$. Then the extended ASE $\hat{Z} \in \mathbb{R}^{n\times D}$ can be partitioned into two parts by $\hat{Z} = \left[\hat{X}|\hat{Y}\right]$, where $\hat{X} \in \mathbb{R}^{n\times d}$ and $\hat{Y} \in \mathbb{R}^{n\times(D-d)}$. The first d dimensions $\hat{X}$ is called the *informative* part, while the remaining dimensions $\hat{Y}$ is called the *redundant* part. If we consider the spectral decomposition of P, which is regarded as the unperturbed version of A, then all of the latent position information is contained in the first d dimensions. This gives the reason for the names.

In the framework of model selection in model-based clustering, both of the informative part and the redundant part need to be parameterized so as to make the posterior probabilities of different models comparable. We will dis-

cuss the model selection procedure in detail in chapter 4. For this purpose, we need to provide a model for the extended ASE with dimension $D(\geq d)$. For this reason, in this section we first present an existing result of asymptotic normality of the rows of adjacency spectral embedding in informative dimensions. We then state a conjecture of a tentative distribution of the embedding in redundant dimensions via simulation observations. The combination of both results establishes the basis of our distributional model for the whole embedding matrix.

3.3.1 Distributional results for ASE in informative dimensions

A remarkable distributional result for the spectral decomposition of random dot product graphs has been proposed [2,106]. In [2], a central limit theorem for the rows of adjacency spectral embedding for RDPG is presented and proven. This result makes it theoretically possible that model-based clustering can be applied for identifying the block memberships in SBM via ASE. In [106], the central limit theorem of ASE is restated in a stronger version, in the sense that its proof does not need an assumption that has been made in [2]. Moreover, the authors have proposed a similar result, namely a central limit theorem, for the rows of Laplacian spectral embedding. By these distributional results, a com-

parison of ASE and LSE for the inference task of clustering is made through Chernoff information, as we have mentioned in the previous section. As the distributional result is essential for model-based clustering, we here present the central limit theorem for ASE. Specifically, since we are more interested in the scenario of random dot product graph with stochastic block model, we interpret the result in this special case:

Theorem 1 (A central limit theorem for ASE)**.** *Let* $\{G^{(n)}\}_{n=1}^{\infty}$ *be a sequence of random graphs, in which each* $G^{(n)} \sim SBM(n, B, \boldsymbol{\pi})$ *is a graph of stochastic block model, with the common positive semidefinite block connectivity probability matrix* $B \in \mathbb{R}^{K\times K}$ *and the prior block probability* $\boldsymbol{\pi} \in \mathbb{R}^{K}$*, as defined in definition 5. Let* $d = rank(B)$ *be the true embedding dimension. Let* $\xi \in \mathbb{R}^{K\times d}$ *be the spectral embedding of* B *with* $B = \xi\xi^T$*.* $\xi_k \in \mathbb{R}^d$*, the* k*th row of* ξ*, denotes the* k*th latent position, one of the* K *possible distinct latent positions in SBM, for* $k = 1, \ldots, K$*. Let random vector* $\boldsymbol{\tau}^{(n)} = (\tau_1^{(n)}, \ldots, \tau_n^{(n)})$ *be the block memberships for the vertices of* $G^{(n)}$*, following the distribution* $\mathbb{P}[\tau_i^{(n)} = k] = \pi_k$ *in the equation (3.5). Let* $X_i^{(n)} = \xi_{\tau_i^{(n)}}$ *be the latent position of the* i*th vertex in* $G^{(n)}$*. Let* $\hat{X}^{(n)} \in \mathbb{R}^{n\times d}$ *be the adjacency spectral embedding of* $G^{(n)}$ *with embedding dimension* d*, defined in definition 6, where* $A = \hat{X}^{(n)}\left(\hat{X}^{(n)}\right)^T$*.* $\hat{X}_i^{(n)}$ *denotes the* i*th row of* $\hat{X}^{(n)}$*. The central limit theorem for the rows of ASE states as follows. For any fixed* $i(\geq 1)$*, there exists a sequence of orthogonal matrices* $\{W^{(n)}\}_{n=1}^{\infty}$

such that for any $x \in \mathbb{R}$*,*

$$\mathbb{P}\left[\sqrt{n}\left(W^{(n)}\hat{X}_i^{(n)} - X_i^{(n)}\right) \leq x \,\middle|\, X_i^{(n)} = \xi_k\right] \longrightarrow \Phi(x; \Sigma_k) \tag{3.12}$$

as $n \longrightarrow \infty$*. In the equation (3.12),* $\Phi(\cdot; \Sigma)$ *denotes the cumulative distribution function for the multivariate normal with mean zero and covariance matrix* Σ*, and* Σ_k *is defined by*

$$\Sigma_k = \mathbb{E}\left[X_0 X_0^T\right]^{-1} \mathbb{E}\left[X_0 X_0^T \left(\xi_k^T X_0 - \xi_k^T X_0 X_0^T \xi_k\right)\right] \mathbb{E}\left[X_0 X_0^T\right]^{-1} \tag{3.13}$$

where X_0 *is a random variable with identical distribution of* $X_i^{(n)}$ *(*$\forall i \leq n$*), i.e. the categorical distribution with* $\mathbb{P}(X_0 = \xi_k) = \pi_k$ *for* $k = 1, \ldots, K$*.*

In (3.12), $W^{(n)}$ is an orthogonal matrix in order to rotate the entire embedding to its corresponding latent positions. The reason for introducing $W^{(n)}$ is due to the non-identifiability of RDPG. We explain it by considering an RDPG G with latent position $X \in \mathbb{R}^{n \times d}$, i.e. $G \sim \text{RDPG}(X)$. The edge probability matrix of G is $P = XX^T$. Let $W \in \mathbb{R}^{d \times d}$ be an arbitrary orthogonal matrix. Since $(XW)(XW)^T = XX^T = P$, the latent position matrix XW will lead to an identical edge probability matrix. So by observing the adjacency matrix A, the task of exact recovery of latent position is inappropriate unless we accept the equivalent class of the latent positions up to rotation. In other words, a rotation of the latent positions will not change its underlying structure. As a result, we

only talk about the consistency of any estimation of the latent positions up to a rotation.

Theorem 1 gives a strong distributional result for adjacency spectral embedding, provided the graph is a random dot product graph with stochastic block model. The theorem states that any row of the ASE follows a multivariate normal distribution around its conditional latent position asymptotically. Considering the latent positions themselves follow an i.i.d categorical distribution into K distinct possible d-dimensional vectors according to B, the unconditioned version of the theorem claims that any row of ASE converges in distribution to a mixture of K multivariate normals, with mixing probabilities $\boldsymbol{\pi}$. That is, for a random dot product graph $G \sim \text{SBM}(n, B, \boldsymbol{\pi})$, the rows of its adjacency spectral embedding with true embedding dimension d are approximately identically distributed as the *Gaussian mixture model* (GMM)

$$f(\cdot; \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k \varphi(\cdot; \xi_k, \Sigma_k) \tag{3.14}$$

for sufficiently large n, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ are the mixing probabilities, $\xi = (\xi_1, \ldots, \xi_K)^T$ are the possible latent positions (up to rotation), and $(\Sigma_1, \ldots, \Sigma_K)$ are the covariance matrices defined in (3.13). Moreover, if $G \sim \text{SBM}(B, \boldsymbol{\tau})$ is an SBM conditioned on the block memberships, then each row of ASE is approximately distributed as a multivariate normal, with mean equal to its

conditioned latent position.

The thoerem gives a complete formula for the covariance matrix of each multivariate normal component, thus fully characterizing the distributional behavior of the rows of ASE around their underlying latent positions. We may utilize it in the subsequent inference task, for example in the task of estimating the block memberships in an SBM [103]. We will also use the result as part of the model structure in the model-based clustering task.

3.3.2 Limiting behavior of ASE in redundant dimensions

One practical issue in applying the results of theorem 1 is that we need to know the true embedding dimension d, which is the rank of the block connectivity probability matrix. As we discussed in section 4.1.1, the true d is unrevealed in real-world applications. Thus in practice, an extended adjacency spectral embedding is performed on a dimension D larger than the true one, followed by a variable selection procedure or by an approach on the entire embedding. In this thesis, our work chooses the latter, namely to model the extended ASE. Let $\hat{Z} = \left[\hat{X}|\hat{Y}\right]$ be the extended ASE with dimension $D(> d)$, where $\hat{X}$ is the informative part and $\hat{Y}$ is the redundant part. Theorem 1 gives the distributional result on $\hat{X}$. We will then present the distributional behavior of $\hat{Y}$.

We have conducted a collection of simulations to get empirical support of the distributional behavior of the redundant part of extended ASE. In the simulation, we generate random graphs according to the stochastic block model SBM(n, B, π), where $B = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.25 \end{bmatrix}$, $\pi = (0.5, 0.5)$ for 2-block graphs, and $B = \begin{bmatrix} 0.2 & 0.1 & 0.08 \\ 0.1 & 0.25 & 0.05 \\ 0.08 & 0.05 & 0.4 \end{bmatrix}$, $\pi = (0.4, 0.4, 0.2)$ for 3-block graphs. The number of vertices of graphs, n, is varied during the simulation. Notice that the true embedding dimension $d = 2$ for 2-block graphs and $d = 3$ for 3-block graphs. We apply the extended adjacency spectral embedding to the adjacency matrix A according to definition 6, but with a fixed dimension $D(> d)$. As defined, the extended ASE $\hat{Z} \in \mathbb{R}^{n \times D}$ is partitioned into an informative part $\hat{X} \in \mathbb{R}^{n \times d}$ and a redundant part $\hat{Y} \in \mathbb{R}^{n \times (D-d)}$ by $\hat{Z} = \left[\hat{X} | \hat{Y}\right]$. $\hat{Y}_i \in \mathbb{R}^{D-d}$ denotes the i-th row of $\hat{Y}$, which corresponds to the i-th vertex with block membership τ_i. The observations of the distributional behavior of $\hat{Y}$ are as follows:

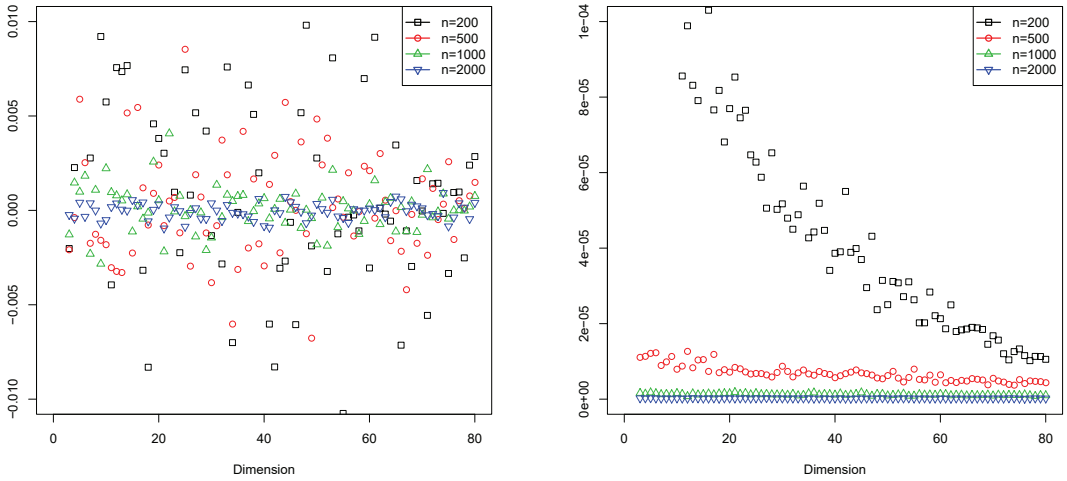
Observation 1: The within-block sample mean of $\hat{Y}_i$ tends to a zero vector as n increases. Figure 3.2 shows the results about the sample mean. In the simulation, the graphs are drawn from the 2-block SBM(n, B, π) with B and π mentioned above. The number of vertices of the graph varies from 200 to 2000, denoted by colors. The extended ASE is applied with dimension $D = 80$. The x-axis shows the indices of the dimension in extended ASE, so the

redundant part $\hat{Y}$ starts from dimension 3 and ends with dimension 80 in this setting. Denoted by $\overline{\hat{Y}}_j^{(1)} \in \mathbb{R}^{D-d}$, the sample mean of the redundant part $\hat{Y}_i$ in dimension $d+j$ for all i in block 1 is calculated by $(j = 1, \ldots, D-d)$

$$\overline{\hat{Y}}_j^{(1)} = \frac{1}{n_1} \sum_{i:\tau_i=1} (\hat{Y}_i)_j \tag{3.15}$$

where n_1 is the number of vertices assigned to block 1, and $(\hat{Y}_i)_j$ is the j-th entry of $\hat{Y}_i$. We plot the sample mean values $\overline{\hat{Y}}^{(1)}$ for each dimension on 1 Monte Carlo trial in figure 3.2a. As n goes large, the points are generally closer to zero. For each dimension, we also plot the mean square errors of $\overline{\hat{Y}}^{(1)}$ from 0 on 100 Monte Carlo replica in figure 3.2b. By observing the results on $n = 200$ and 500, we see that the sample mean gets smaller on larger dimensions. It has also been shown that $\overline{\hat{Y}}^{(1)}$ is approaching a zero vector as n goes large. For example, when $n = 2000$, the mean square errors are almost exactly 0. We conclude that the within-block sample mean of $\hat{Y}_i$ tends to a zero vector as n increases.

Observation 2: The within-block sample variance of each dimension of $\hat{Y}_i$ tends to a constant for large n. Figure 3.3 shows the results about the sample variance. In the simulation, the graphs are again drawn from the 2-block SBM(n, B, π) with B and π mentioned above. The number of vertices of the graph varies from 200 to 2000. The extended ASE is applied with dimension $D = 80$. The x-axis shows the indices of the dimension in extended ASE. So the



(a) Sample mean values

(b) Mean square errors from 0

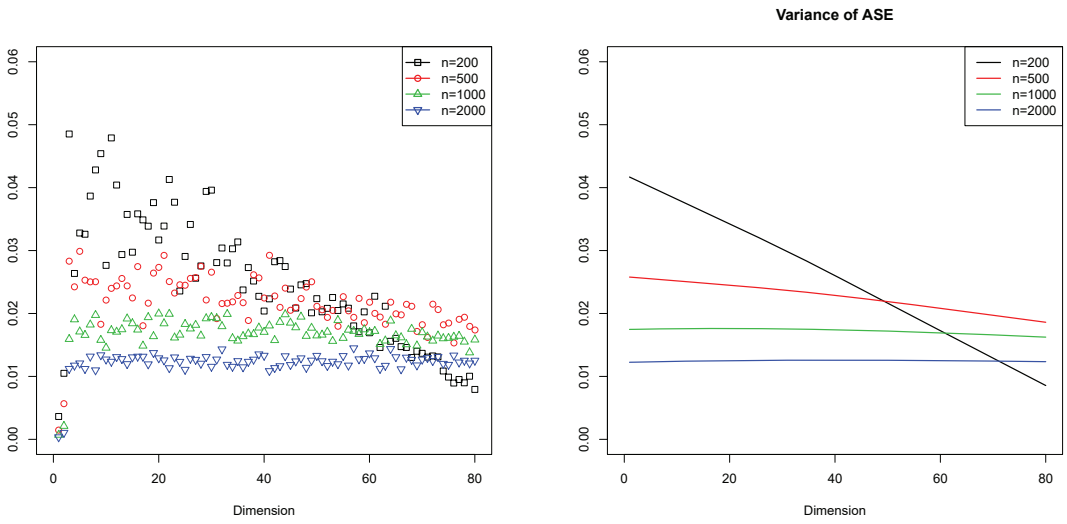
Figure 3.2: The sample mean of the redundant part $\hat{Y}_i$ for all i in block 1. The graphs are drawn from the 2-block SBM(n, B, π) with given B and π. The number of vertices of the graph varies from 200 to 2000, denoted by colors. The extended ASE is applied with dimension $D = 80$. The sample mean is calculated from dimension 3 to dimension 80. The x-axis indicates the indices of the dimension in extended ASE. (a) Sample mean values on 1 Monte Carlo trial. (b) Mean square errors of the sample mean from 0 on 100 Monte Carlo replica.

redundant part $\hat{Y}$ starts from dimension 3 and ends with dimension 80 in this setting. For each dimension j, the sample variance of $(\hat{Y}_i)_j$ for all i in block-1 is calculated by

$$s_j^2 = \frac{1}{n_1 - 1} \sum_{i:\tau_i=1} \left((\hat{Y}_i)_j - \overline{\hat{Y}}_j^{(1)} \right)^2 \tag{3.16}$$

where n_1 is the number of vertices assigned to block 1 and $\overline{\hat{Y}}_j^{(1)}$ is the corresponding sample mean in block-1. We plot the sample variance values s_j^2 for each dimension on 1 Monte Carlo trial in figure 3.3a. The points on the first

two dimensions behave like outliers, because they are sample variance from the informative part. To have a clearer view, we fit a curve by LOWESS smoother on the means of sample variance over 20 Monte Carlo replica, in figure 3.3b. We observe that the within-block sample variance tends to get smaller for larger dimensions. Moreover, the decreasing rate of the values over different dimensions tends to be smaller as n increases. When $n = 2000$, the sample variances are almost equal. So we conclude that the within-block sample variance of each dimension of $\hat{Y}_i$ tends to a constant for large n.



(a) Sample variance values

(b) Curves fitted by LOWESS smoother

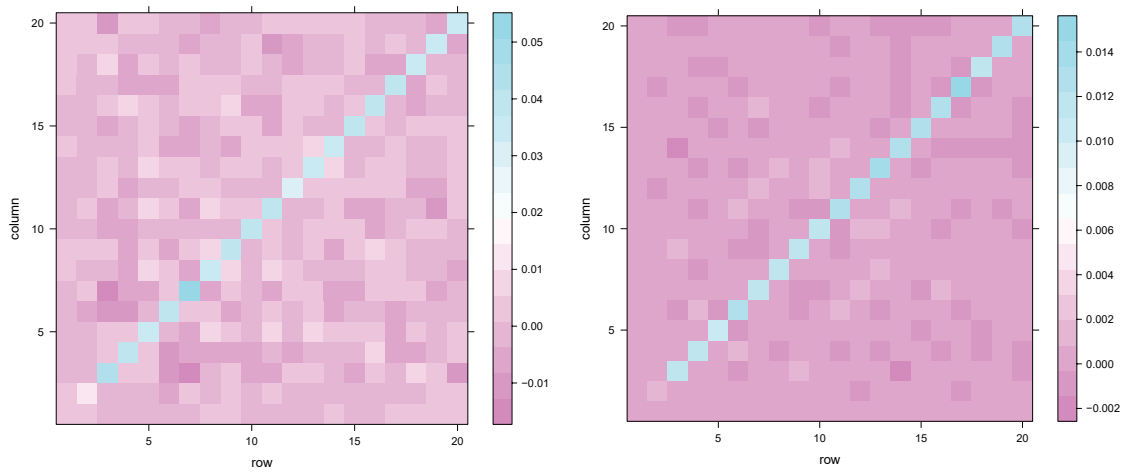
Figure 3.3: The sample variance of each dimension of $\hat{Z}_i$ for all i in block 1. The graphs are drawn from the 2-block SBM(n, B, π) with given B and π. The number of vertices of the graph vaires from 200 to 2000, denoted by colors. The extended ASE is applied with dimension $D = 80$. The sample variance is calculated from dimension 1 to dimension 80. The x-axis indicates the indices of the dimension in extended ASE. (a) Sample variance values on 1 Monte Carlo trial. (b) Curves of the sample variance values against dimensions fitted by a LOWESS smoother on 20 Monte Carlo replica.

Observation 3: The within-block sample covariance matrix of $\hat{Y}_i$ tends to be diagonal, and the covariance between informative and redundant dimensions tend to be zero, for large n. Figure 3.4 shows the results about the sample covariance matrix. In the simulation, the graphs are again drawn from the 2-block SBM(n, B, π) with B and π mentioned above. The number of vertices of the graph is set to be 200 and 2000. The extended ASE is applied with dimension $D = 20$. The x-axis and y-axis indicate the indices of the dimensions in extended ASE, respectively. So the redundant part $\hat{Y}$ starts from dimension 3 and ends with dimension 20 in this setting. The sample covariance matrix of $\hat{Z}_i$ for all i in block-1 is calculated by

$$\Sigma^{(1)} = \frac{1}{n_1 - 1} \sum_{i:\tau_i=1} \left(\hat{Z}_i - \overline{\hat{Z}}^{(1)} \right) \left(\hat{Z}_i - \overline{\hat{Z}}^{(1)} \right)^T \tag{3.17}$$

where n_1 is the number of vertices assigned to block 1, $\hat{Z}_i \in \mathbb{R}^{D\times 1}$ is the i-th row of extended ASE (but regarded as a column vector), and $\overline{\hat{Z}}^{(1)} \in \mathbb{R}^{D\times 1}$ is corresponding sample mean in block 1. We plot the sample covariance matrix $\Sigma^{(1)}$ for $n = 200$ in figure 3.3a and $n = 2000$ in figure 3.3b. Values are shown in different colors. The matrix contains both informative dimensions and redundant dimensions. We observe that the diagonal values in the matrix of redundant dimensions concentrates on a constant for $n = 2000$, which is consistent with the result shown in figure 3.3. The off-diagonal values in the matrix of redundant

dimensions tend to be zero as n increases. For $n = 2000$, the covariance matrix presents a block diagonal structure, partitioned by the true embedding dimension d. So we conclude that the within-block sample covariance matrix of $\hat{Y}_i$ tends to be diagonal for large n. Moreover, the covariance between informative and redundant dimensions tend to be zero, for large n.

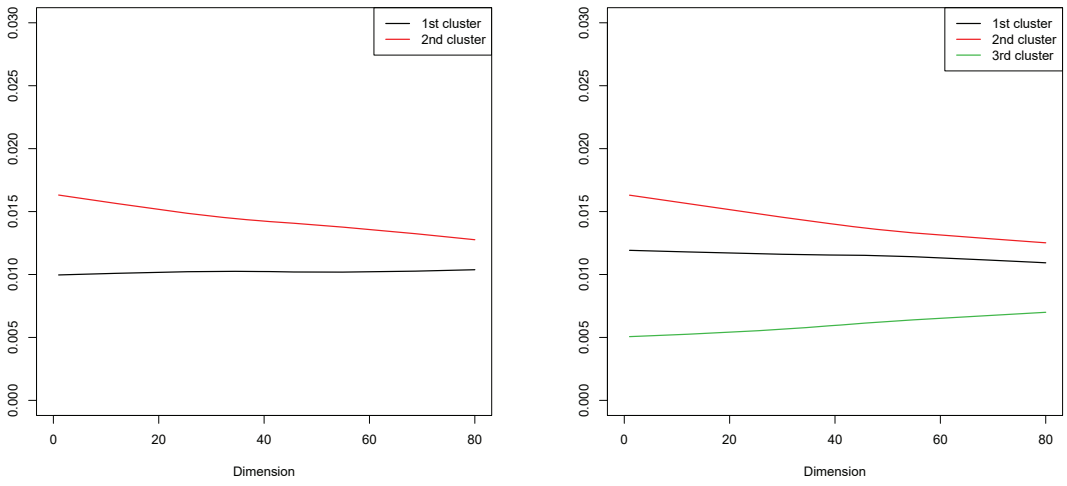


(a) n=200 (b) n=2000

Figure 3.4: The sample covariance matrix of $\hat{Z}_i$ for all i in block 1. The graphs are drawn from the 2-block SBM(n, B, π) with given B and π. The extended ASE is applied with dimension $D = 20$. The x-axis and y-axis indicate the indices of the dimensions in extended ASE, respectively. Values are shown in different colors. (a) $n = 200$; (b) $n = 2000$.

Observation 4: The within-block sample variances are distinct for different blocks. Figure 3.5 shows the results about the sample variance for different blocks. Graphs are drawn from a 2-block SBM(n, B, π) in figure 3.5a and from a 3-block SBM(n, B, π) in figure 3.5b, with B and π mentioned above.

The number of vertices of the graph is 3000. The extended ASE is applied with dimension $D = 80$. The x-axis shows the indices of the dimension in extended ASE. For each dimension j, we calculate the sample variance of $(\hat{Y}_i)_j$ respectively for i in the different blocks. So for an SBM with K blocks, we have K sample variances. We plot the sample variance values s_j^2 for each dimension. Curves are fitted by LOWESS smoother. We observe that the sample variance from different blocks are different, both for the 2-block graph and the 3-block graph. So we conclude that the within-block sample variances are distinct for different blocks.



(a) 2-block graph

(b) 3-block graph

Figure 3.5: The within-block sample variances of $\hat{Y}_i$. Colors indicates different blocks. The graphs are drawn from (a) 2-block SBM(n, B, π) and (b) 3-block SBM(n, B, π) with given B and π. Number of vertices $n = 3000$. The extended ASE is applied with dimension $D = 80$. The sample variance is calculated from dimension 3 to dimension 80. The x-axis indicates the indices of the dimension in extended ASE.

3.4 Probability models for extended adjacency spectral embedding

For the extended adjacency spectral embedding of a graph with stochastic block model, the distributional result for its informative part is presented in section 3.3.1, and the limiting behavior of its redundant part is shown in section 3.3.2. There are few theoretical results about both the redundant part and the whole extended ASE in the literature. However, a distributional model is important and necessary for variable selection followed by any inference task. In this section, we provide a finite mixture model for the extended ASE with SBM. Although it has not been proven analytically at this point, we believe the model is asymptotically close to the truth, both by our observations on the limiting behavior and by the performance of the subsequent inference task based on this model.

We first state our conjectures about the distribution which the redundant part of extended ASE follows. We consider random dot product graphs with a K-block stochastic block model. Again, let the extended ASE $\hat{Z} \in \mathbb{R}^{n \times D}$ be partitioned into informative part $\hat{X} \in \mathbb{R}^{n \times d}$ and redundant part $\hat{Y} \in \mathbb{R}^{n \times (D-d)}$ by $\hat{Z} = \left[\hat{X} | \hat{Y}\right]$. d denotes the true dimension of latent positions, and D denotes the actual embedding dimension. Based on the simulations conducted on $\hat{Y}$ in section 3.3.2, our conjecture is as follows: Any row of $\hat{Y}$ is asymptotically

multivariate Gaussian distributed conditioned on its block membership. That is, for any $i \in [n]$,

$$\hat{Y}_i | \tau_i = k \longrightarrow N(\mu_k, \Sigma_k) \tag{3.18}$$

approximately if n is sufficiently large. If we consider the sample statistics in the simulation to be a good estimation of the Gaussian parameters, we can further specify the model. By observation 1, the within-block sample mean of $\hat{Y}_i$ tends to a zero vector as n increases. This implies we may assume $\mu_k = 0$ for all $k \in [K]$. By observation 3, the within-block sample covariance matrix of $\hat{Y}_i$ tends to be diagonal for large n, so Σ_k is approximately a diagonal matrix. By observation 2, the within-block sample variance of each dimension of $\hat{Y}_i$ tends to a constant for large n, so the diagonal entries of Σ_k can be a common parameter. Together with the diagonal assumption, we can assume $\Sigma_k = \alpha_k I$, where I is the identity matrix. Finally by observation 4, the within-block sample variances are distinct for different blocks. This inspires us to assume different α_k if the conditioned block membership is different. Therefore our conjecture becomes

$$\hat{Y}_i | \tau_i = k \longrightarrow N(0, \alpha_k I) \tag{3.19}$$

By combining the conjecture of the redundant dimensions with the theoretical results for the informative dimensions (see section 3.3.1), we propose a *Gaussian mixture model* (GMM) on the extended ASE $\hat{Z}$ as follows:

Model 1 (GMM for extended ASE of undirected graphs)**.** *Let*

$$f(\cdot;\theta(d,K)) = \sum_{k=1}^{K} \pi^{(k)} \varphi(\cdot;\mu^{(k)},\Sigma^{(k)}) \tag{3.20}$$

be a family of density functions for a D dimensional GMM random vector, where $\{\pi^{(k)}\}_{k=1}^{K}$ are the mixing probabilities, $\{\mu^{(k)}\}_{k=1}^{K}$ are the mean vectors, and $\{\Sigma^{(k)}\}_{k=1}^{K}$ are the covariance matrices. Furthermore, they satisfy

$$\sum_{k=1}^{K} \pi^{(k)} = 1 \tag{3.21}$$

$$\mu^{(k)} = [\mu_1^{(k)},\ldots,\mu_d^{(k)},0,\ldots,0]^T \tag{3.22}$$

and

$$\Sigma^{(k)} = \begin{bmatrix} \tilde{\Sigma}^{(k)} & 0 \\ 0 & \sigma^{2^{(k)}} I \end{bmatrix} \tag{3.23}$$

where $\tilde{\Sigma}^{(k)}$ is a $d\times d$ positive semidefinite matrix, and I is a $(D-d)\times(D-d)$ identity matrix. In this notation, $\theta(d,K)$ denotes the parameters $\{\rho^{(k)},\mu^{(k)},\Sigma^{(k)}\}_{k=1}^{K}$, specifically $\theta(d,K) = \left\{\pi^{(k)},\left[\mu_1^{(k)},\ldots,\mu_d^{(k)}\right],\tilde{\Sigma}^{(k)},\sigma^{2^{(k)}}\right\}_{k=1}^{K}$, which belongs to the parameter space $\Theta(d,K)$.

We establish our probability model for the extended adjacency spectral embedding of $G \sim \text{SBM}(n,B,\boldsymbol{\pi})$. Let the extended ASE be $\hat{Z} \in \mathbb{R}^{n\times D}$, then our

conjecture states, for any $i \in [n]$,

$$\hat{Z}_i \sim f(\cdot; \theta^*(d_0, K_0)) \tag{3.24}$$

approximately for sufficiently large n, where $f(\cdot; \theta(d, K))$ is the density function defined in Model 1, d_0 is the true dimension of latent position, K_0 is the true number of blocks, and $\theta^*(d, K)$ is the true underlying parameters of the GMM. This conjecture states that the rows of extended ASE are identically distributed as the our GMM, but we haven't assumed that they are independent. In fact, it has been shown that the rows of ASE are not independent [2, 106]. However, for ease of analysis we will proceed in the consistency theorem and in the calculation of BICs by ignoring dependency, because the later experimental results show that the independent assumption is tractable and acceptable.

Chapter 4

Simultaneous Model Selection and Vertex Clustering

Consider the vertex clustering problems on graphs with stochastic block model. From chapter 3, we have seen that the adjacency spectral embedding $\hat{X}$ of a graph with SBM contains a noisy version of the clustering information which comes from the perfect version in latent position matrix X. The theoretical support for the effectiveness of estimating X by $\hat{X}$ can be found from either the consistency results [66,101,102] or the distributional results [2,106]. However, these results are all based on the assumption that true embedding dimension d is known and ASE is applied with that dimension, yet in real-world applications d is unknown. Thus in order to refer the theoretical results in the subsequent inference task, a dimension selection procedure is usually

necessary. Moreover, even if we can apply ASE on an accurate dimension with a robust estimation technique, most of the clustering algorithms rely on the estimation of number of clusters. In the framework of model-based clustering, we refer to them as the model selection problems.

In this thesis, we are interested in the model selection problem under the specific task of clustering the vertices of a SBM graph by its adjacency spectral embedding. In this task the traditional data points in the Euclidean space are the rows of ASE. The model refers to the distributional assumption for those data points. If we restrain our model in the Gaussian mixture model family, as we have explained in section 3.4, our goal of model selection is to determine the structure of the free parameters.

In this chapter, we first formally define the model selection problem under the framework of vertex clustering via adjacency spectral embedding. We then present some important consistency results of the estimation of the two model parameters discussed above, namely the embedding dimension d and the number of components K. We will see all the existing methods treat the two model parameters separately. We propose a framework in which the two model parameters are considered simultaneously in our model. A theoretical result showing the consistency of our estimation approach will be presented and proven. The theorem ensures that our estimators of d and K converge to underlying truth for sufficiently large graphs. Inspired by the framework and

theorem of simultaneous model selection, we conclude the chapter by introducing two efficient algorithms of vertex clustering followed by detailed discussion of implementation.

4.1 Principled methods for consecutive model selection

There are two model selection problems in spectral clustering in general as we discussed above. Particularly, for our task to cluster the rows of adjacency spectral embedding of a graph with stochastic block model by model-based clustering, one problem is to estimate the dimension of the latent vector d, while the other is to estimate the number of blocks K. We refer to the estimate of d as the actual embedding dimension, denoted by $\hat{d}$, and to the estimate of K as the mixture complexity of the model, denoted by $\hat{K}$.

The traditional solution to these two model selection problems is usually conducted in a successive procedure, namely applying variable selection or dimension reduction technique to estimate $\hat{d}$ first, then applying model selection technique on the data with $\hat{d}$ dimensional adjacency spectral embedding to estimate $\hat{K}$. To be specific, again let $G \sim \text{SBM}(n, B, \boldsymbol{\pi})$ be the graph of interest with stochastic block model. Let $A = \hat{U}\hat{\Lambda}\hat{U}^T$ be the spectral decomposition of the adjacency matrix $A \in \mathbb{R}^{n\times n}$ defined by (3.7). We first apply extended adja-

cency spectral embedding on A with dimension D which is a loose upper bound of d. Let the result $\hat{Z} = \hat{U}_{[D]}\hat{\Lambda}_{[D]}^{\frac{1}{2}}$ be the extended ASE defined by (3.11). The first model selection procedure is to determine the dimension of the Euclidean space in which the underlying clustering structure of the embedded Euclidean data is shown clearly and efficiently. The dimension $\hat{d}$ is usually estimated by some algorithm which uses the spectral information contained in $\hat{\Lambda}$. Then we truncate the extended ASE up to $\hat{d}$ dimension, denoted by $\hat{Z}_{[\hat{d}]} = \hat{U}_{[\hat{d}]}\hat{\Lambda}_{[\hat{d}]}^{\frac{1}{2}}$. This truncated ASE $\hat{Z}_{[\hat{d}]}$ is the new data matrix whose rows are to be clustered. The second model selection procedure is to determine the mixture complexity of the model from which the rows of $\hat{Z}_{[\hat{d}]}$ are supposed to be generated. The number of components $\hat{K}$ is usually estimated by some algorithm which uses the clustering structure of $\hat{Z}_{[\hat{d}]}$. As long as $\hat{d}$ and $\hat{K}$ are obtained, the parameter space of the finite mixture model has been determined. Therefore we are ready to conduct the subsequent model-based clustering on $\hat{Z}_{[\hat{d}]}$ to finalize the inference task of vertex clustering. Since the two model selection procedures are executed in sequence, we name this framework *consecutive model selection* (CMS). A general framework of CMS is summarized in algorithm 1.

4.1.1 Choice of embedding dimension

We have discussed the general variable selection problems in section 2.4, where we refer to a vast collection of literature. Here we focus on the sce-

Algorithm 1 Framework of consecutive model selection (CMS)

Input: The adjacency matrix $A \in \mathbb{R}^{n \times n}$; an upper bound D of the embedding dimension.

1: **procedure** CMS(A, D)

2: Perform spectral decomposition by $A = \hat{U}_{[D]} \hat{\Lambda}_{[D]} \hat{U}_{[D]}^T$

3: Estimate $\hat{d}$ by $\hat{\Lambda}_{[D]}$

4: Compute $\hat{Z}_{[\hat{d}]} = \hat{U}_{[\hat{d}]} \hat{\Lambda}_{[\hat{d}]}^{\frac{1}{2}}$

5: Estimate $\hat{K}$ by $\hat{Z}_{[\hat{d}]}$

6: **end procedure**

Output: A $\hat{d}$-dimensional mixture model with $\hat{K}$ components.

nario of choosing the embedding dimension $\hat{d}$ for adjacency spectral embedding. As we mentioned in section 2.4, there are no best methods for this task. We pick the so-called *scree plot* method for the dimension selection task in the consecutive model selection framework as a comparison, because it is one of the most commonly used and easily implemented approaches. As a singular value thresholding (SVT) approach, the idea is to plot the eigenvalues or singular values in descending order and then find the "gap" which divides the eigen/singular values into a signal part and a noise part. This is based on the assumption that eigen/singular values of signal dimensions are well separated in magnitude from that of noise dimensions. However, it is highly subjective to decide where the "gap" or "elbow" is; for this reason we here present an ef-

fective method, which has been proposed in [115], to identify the elbow in the scree plot.

The main idea of the method is to maximize a profile-likelihood function over the position of the elbow. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$ be the ordered eigenvalues (or other measures of importance) obtained from the data. Provided there is an elbow between position d and $d+1$, the author assumes that $\{\lambda_1, \ldots, \lambda_d\}$ and $\{\lambda_{d+1}, \ldots, \lambda_D\}$ are samples from two different distributions, with probability density function $f(\cdot; \theta_1)$ and $f(\cdot; \theta_2)$ respectively. With the independence assumption, the author provides a profile log-likelihood function as

$$l(d) = \sum_{i=1}^{d} \log f(\lambda_i; \hat{\theta}_1(d)) + \sum_{i=d+1}^{D} \log f(\lambda_i; \hat{\theta}_2(d)) \tag{4.1}$$

where $\hat{\theta}_1(d)$ and $\hat{\theta}_2(d)$ are the maximum likelihood estimators (MLE) for θ_1 and θ_2. Then the estimation of the elbow can be obtained by maximizing the profile function, i.e.

$$\hat{d} = \arg\max_{d} l(d) \tag{4.2}$$

In practice, Gaussian distribution is used for $f(\cdot; \theta_1)$ and $f(\cdot; \theta_2)$, that is

$$f(\lambda; \theta_k) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\lambda - \mu_k)^2}{2\sigma^2}\right) \tag{4.3}$$

for $k = 1, 2$. The author assumes that two Gaussian distribution have different

mean μ_1 and μ_2, but have the same standard deviation σ. This is because overly flexible models may cause the profile log-likelihood function unbounded. Given the elbow position d, the MLEs are given by

$$\hat{\mu_1} = \frac{1}{d}\sum_{i=1}^{d}\lambda_i \tag{4.4}$$

$$\hat{\mu_2} = \frac{1}{D-d}\sum_{i=d+1}^{D}\lambda_i \tag{4.5}$$

$$\hat{\sigma}^2 = \frac{1}{D-2}\left[\sum_{i=1}^{d}(\lambda_i - \hat{\mu_1})^2 + \sum_{i=d+1}^{D}(\lambda_i - \hat{\mu_2})^2\right] \tag{4.6}$$

By pluging-in the MLEs in equation (4.1), we can compute the profile log-likelihood function for $d = 1, \ldots, D-1$. Then the elbow $\hat{d}$ will be identified by maximizing the profile likelihood function in (4.2). We refer to this scree plot algorithm as "ZG" in honor of the authors.

However, in practice the elbow estimated by ZG algorithm sometimes underestimates the signal dimension for finite samples. To illustrate this phenomenon, we assume the eigenvalues in descending order are $(\lambda_1, \ldots, \lambda_4) = (2, 1, 0.2, 0.1)$. Then estimation of the elbow given by ZG algorithm is $\hat{d} = 1$, which means only the first dimension is considered as signal. But just by observing the scree plot, it is very possible that the second largest eigenvalue 1 is also in the signal dimension. This example illustrates that ZG may not be suitable in the case where signal eigenvalues are relatively far apart from each other. Moreover, the optimal dimension for the subsequent inference task may

not always be exactly the true one due to bias-variance tradeoff [44]. For this purpose, we provide an extension of the ZG algorithm to make it more flexible in the sense that multiple elbows in the scree plot are detected. To be specific, we first apply ZG algorithm on all the eigenvalues, say $(\lambda_1, \ldots, \lambda_D)$, and get the first elbow, denoted by $\hat{d}_1$. Then we apply again the ZG algorithm on the eigenvalues beyond the $\hat{d}_1$ dimension, say $(\lambda_{\hat{d}_1+1}, \ldots, \lambda_D)$, and get the second elbow, denoted by $\hat{d}_2 (> \hat{d}_1)$. This elbow could still be a remarkable "gap" if we only see the scree plot after dimension $\hat{d}_1$. Similarly, we can keep applying ZG algorithm in this way until the last elbow appears at the position D. Thus we will have a sequence of elbows which capture all the "gaps" in the scree plot. In applications, we often find that the second or even the third elbow is more plausible to be close to the true dimension, as we will show in Section 4.4 and Chapter 5. The extension of the ZG algorithm is summarized in algorithm 2.

The extended ZG algorithm can detect multiple elbows in the scree plot, which increases the robustness of the algorithm in the sense that the true dimension is probably closer to one of the other elbows than the first one (in the original ZG algorithm). However, the extended version still suffers from the drawback that determining which elbow to be the optimal one is empirical. That is, an ordinal number l is required when applying the extended ZG algorithm to estimate the embedding dimension. The choice of the number l indicates that we choose the l-th elbow from the ZG algorithm as our embed-

ding dimension. We will see later that none of the specific elbows dominate the others in all cases. In real-world applications, this issue becomes more serious as we usually have no prior knowledge of the ground truth. Despite this, ZG algorithm is still one of the best scree plot methods. We will use it in the consecutive model selection procedure of our problem in competition.

Algorithm 2 Extended elbow detection algorithm by profile likelihood

Input: The measure of importance $(\lambda_1, \ldots, \lambda_D)$ in descending order.

function ZG($\lambda_1, \ldots, \lambda_D$)
  for $d \leftarrow 1 : D$ **do**
    Compute MLEs by (4.4 - 4.6)
    Compute profile likelihood $l(d)$ by (4.1)
  end for
  Get the first elbow $\hat{d}_1 \leftarrow \arg\max_d l(d)$
  if $\hat{d}_1 < D$ **then**
    $(\hat{d}_2, \ldots, \hat{d}_m) \leftarrow$ ZG($\lambda_{\hat{d}_1+1}, \ldots, \lambda_D$)
  end if
  return $(\hat{d}_1, \ldots, \hat{d}_m)$
end function

Output: A sequence of elbows $(\hat{d}_1, \ldots, \hat{d}_m)$.

4.1.2 Choice of mixture complexity

By the estimate of embedding dimension $\hat{d}$, we now have the truncated ASE $\hat{Z}_{[\hat{d}]}$, whose rows are considered to be the data points on which clustering procedure is applied. Now we face the second model selection problem, namely determining the number of clusters. We have already discussed the general problem of choosing the number of groups, for example in k-means, in section 2.3. As we mentioned in theorem 1, the rows of adjacency spectral embedding with true embedding dimension are identically distributed as a Gaussian mixture model. Thus we here focus on the scenario where model-based clustering, specifically Gaussian mixture model, is used on the data $\hat{Z}_{[\hat{d}]}$.

A ubiquitous approach for determining the mixture complexity is by comparing an information criterion, usually a penalized likelihood function. We have referred to several well-known information criteria in section 2.3. The Bayesian information criterion (BIC) is probably the most well-studied one among them. It has been proven that estimating the mixture complexity of a Gaussian mixture model by BIC is consistent [49]. Even for finite data many applications show that BIC performs well in the model selection tasks [14, 21, 90, 98], so we choose to use BIC method in our second model selection problem.

The formula of *Bayesian information criterion* (BIC) is given in [94]

$$\mathbf{BIC}(X|M) = 2\log P_M(X|\hat{\theta}) - \eta_M \log n \tag{4.7}$$

where M is the model of interest, $P_M(\cdot|\theta)$ is the likelihood function of M, $\hat{\theta}$ is the maximum likelihood estimators of the parameters, η_M is the number of free parameters in M, X is the observed data, and n is the number of observations. Let the probability function of the Gaussian mixture model be formulated as

$$f(\cdot;\theta(G)) = \sum_{k=1}^{G} \pi_k \varphi(\cdot;\mu_k,\Sigma_k) \tag{4.8}$$

where $\theta(G)$ is the set of free parameters in the GMM with G components. The approach of choosing the mixture complexity by BIC is simply by comparing the BIC values evaluated on GMM with different numbers of components. The estimator $\hat{K}$ is chosen from the GMM with the largest BIC value, i.e.

$$\hat{K} = \arg\max_{G} \mathbf{BIC}(X|M_G) \tag{4.9}$$

where M_G is a finite mixture model with G components. Usually, we empirically set a number $K_{\max}$ as an upper bound of the possible mixture complexity, and evaluate the BIC values on $M_1,\ldots,M_{K_{\max}}$ in (4.9). If the data points in X are independently identically distributed as the GMM defined in (4.8), then the

log-likelihood function of M_G in (4.7) can be calculated as

$$\log P_{M_G}(X|\hat{\theta}(G)) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{G} \hat{\pi}_k \varphi(X_i; \hat{\mu}_k, \hat{\Sigma}_k) \right] \tag{4.10}$$

where $\hat{\theta}(G) = \{\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k\}_{k=1}^{G}$ are the MLEs and X_i is the i-th row of X. It is usually impractical to obtain the closed-form solutions of MLEs, defined in equation (2.2), due to the complicated and multi-modal form of the joint likelihood. The standard approach to compute the MLEs in the finite mixture model is the expectation maximization (EM) algorithm [23, 70]. We will discuss more details of EM algorithm in section 4.3. The BIC method to choose the mixture complexity is summarized in algorithm 3.

By applying algorithm 2 and algorithm 3 sequentially, we can now embody the framework of consecutive model selection stated in algorithm 1. To finalize the vertex clustering algorithm, we apply model-based clustering on $\hat{Z}_{[\hat{d}]}$ with a $\hat{K}$-component GMM. As we have discussed in section 2.2, each data point is clustered by the *maximum a posteriori* (MAP) rule, i.e. the i-th data point is assigned to the $\hat{\tau}_i$-th component by

$$\hat{\tau}_i = \arg\max_j \{\hat{z}_{ij}\} \tag{4.11}$$

Algorithm 3 Choosing mixture complexity of GMM by BIC

Input: Matrix X whose rows comprise the n data points; an upper bound $K_{\max}$ of number of clusters.

1: **function** BIC($X, K_{\max}$)
2: **for** $G \leftarrow 1 : K_{\max}$ **do**
3: Compute MLEs in (4.10) by EM algorithm
4: Compute BIC$(X|M_G)$ by (4.10) and (4.7)
5: **end for**
6: $\hat{K} = \arg\max_G \text{BIC}(X|M_G)$
7: **return** $\hat{K}$
8: **end function**

Output: An estimate of number of components $\hat{K}$.

where

$$\hat{z}_{ij} = \frac{\hat{\pi}_j \varphi(X_i; \hat{\mu}_j, \hat{\Sigma}_j)}{\sum_{k=1}^{\hat{K}} \hat{\pi}_k \varphi(X_i; \hat{\mu}_k, \hat{\Sigma}_k)} \tag{4.12}$$

is the posterior probability that the i-th data point belongs to the j-th component. We summarize the model-based clustering method in algorithm 4. Finally, the whole procedure of vertex clustering via CMS is shown in algorithm 5. We call the method BIC $\circ$ ZG.

Algorithm 4 Model-based clustering via GMM

Input: X whose rows comprise the n data points; mixture complexity K.

function GMM(X, K)
  for $G \leftarrow 1, \ldots, K$ **do**
    Compute MLEs in (4.10) by EM algorithm
  end for
  for $i \leftarrow 1 : n$ **do**
    for $j \leftarrow 1 : K$ **do**
      $\hat{z}_{ij} \leftarrow \frac{\hat{\pi}_j \varphi(X_i; \hat{\mu}_j, \hat{\Sigma}_j)}{\sum_{k=1}^{K} \hat{\pi}_k \varphi(X_i; \hat{\mu}_k, \hat{\Sigma}_k)}$
    end for
    $\hat{\tau}_i \leftarrow \arg\max_j \{\hat{z}_{ij}\}$
  end for
end function

Output: Clustering label $(\hat{\tau}_1, \ldots, \hat{\tau}_n)$.

4.2 Approaches of simultaneous model selection

The consecutive model selection approaches defined in algorithm 1 suffer from three drawbacks. The first drawback is that there are no best methods in the procedure of selection of embedding dimension, as we have discussed in section 3.2.2. Even for the ZG algorithm (defined in algorithm 2), one of the

Algorithm 5 Vertex clustering via consecutive model selection (CMS)

Input: The adjacency matrix $A \in \mathbb{R}^{n \times n}$; an upper bound D of embedding dimension; an upper bound $K_{\max}$ of mixture complexity; an ordinal number l for the elbow.

1: **function** BIC $\circ$ ZG($A, D, K_{\max}, l$)

2: Perform spectral decomposition by $A = \hat{U}_{[D]}\hat{\Lambda}_{[D]}\hat{U}_{[D]}^T$

3: $\hat{d} \leftarrow \left[\text{ZG}(\text{diag}(\hat{\Lambda}_{[D]}))\right]_l$ by algorithm 2

4: $\hat{Z}_{[\hat{d}]} \leftarrow \hat{U}_{[\hat{d}]}\hat{\Lambda}_{[\hat{d}]}^{\frac{1}{2}}$

5: $\hat{K} \leftarrow \text{BIC}(\hat{Z}_{[\hat{d}]})$ by algorithm 3

6: $(\hat{\tau}_1, \ldots, \hat{\tau}_n) \leftarrow \text{GMM}(\hat{Z}_{[\hat{d}]}, \hat{K})$ by algorithm 4

7: **end function**

Output: The clustering label $(\hat{\tau}_1, \ldots, \hat{\tau}_n)$.

best scree plot methods, the choice of the ordinal number l of the elbow is still a subjective job and as such is not reliable. The second drawback is that the latter model selection procedure, namely selection of mixture complexity, completely depends on the result of the former model selection procedure, namely selection of embedding dimension. This is because the methods of estimating mixture complexity use the structure of the truncated data with estimated embedding dimension. The error from the former one may accumulate and thus highly affect the outcome of the latter one, even if the methodology of the latter, for example algorithm 3, is reliable. The third drawback is that after choosing

the embedding dimension the data is cut off so that any variables beyond the embedding dimension $\hat{d}$ is thrown away. Although clustering task could be degraded by considering unnecessary variables, the accuracy of the estimation in the former step is not assured. Moreover, even if the estimate $\hat{d}$ is close to the true one, it is possible that there is useful information contained in the redundant dimension for either clustering or estimating the mixture complexity, especially on finite data. An example of this is the phenomenon discussed in section 3.3.2.

To overcome the shortcoming of consecutive model selection, we propose a framework of model selection and subsequent vertex clustering in which the redundant part of the extended adjacency spectral embedding is taken into account. The work is inspired by the variable selection framework proposed in [88], which we have discussed in section 2.4.2. In contrast with the consecutive model selection procedure, our approach selects the embedding dimension, mixture complexity and the clustering model simultaneously. For this reason we name this framework *simultaneous model selection* (SMS). Within this framework, we focus on the procedure specifically tailored for vertex clustering tasks on the graph with stochastic block model. In this section, we will cast the framework of SMS followed by a theoretical result on the consistency of the model parameter estimates.

4.2.1 Motivation and framework

The idea of simultaneous model selection comes from the basis of model comparison in [88]. In general, let M_1 and M_2 be the two models that we are going to compare, and X be the observations. Intuitively, the X is more likely generated from the model with higher posterior probability, where the posterior probability of M_i is defined as $P(M_i|X)$ for $i = 1, 2$. It is clear that there is an underlying assumption: M_1 and M_2 are both distributional models which characterize the same random vector, with X as its realization. This implies that M_1 and M_2 have the same dimensionality. In other words, models for different random vectors are not comparable. In the example of adjacency spectral embedding, one cannot compare two models M_1 and M_2, if M_1 is for ASE with embedding dimension d_1 and M_2 is for ASE with a different embedding dimension d_2. Therefore, in the framework of consecutive model selection, model comparison is not applicable because the embedded data has different dimensions. In order to compare various models, we need to fix the dimension of the data. This inspired us to establish a family of the models which describe the extended adjacency spectral embedding with a fixed dimension D.

Assume now M_1 and M_2 are models that both describe the same random vector. By Bayes' theorem, we notice that the posterior probability of the model is proportional to the product of the prior and the integrated likelihood, i.e. for

$i = 1, 2$

$$P(M_i|X) \propto P(M_i)P(X|M_i) \tag{4.13}$$

where we call $P(X|M_i)$ the *integrated likelihood* because it can be obtained by integrating over all the unknown parameters in the model, i.e.

$$P(X|M_i) = \int P(X|\theta_i, M_i)P(\theta_i|M_i)d\theta_i \tag{4.14}$$

Since usually we assume no preference between the two models, we can ignore the prior probability $P(M_i)$ term and just compare the integrated likelihoods. Thus the criterion we use is the Bayes factor, which is defined as the ratio of the integrated likelihoods

$$B_{12} = P(X|M_1)/P(X|M_2) \tag{4.15}$$

We say M_1 is in favor if $B_{12} > 1$. However, as we have discussed in section 2.3, computing the integrated likelihood is impractical. Alternatively, Bayesian information criterion, defined in (4.7), has been shown to be a good approximation of the integrated likelihood. Consequently, we say the observation is in favor of the model with higher BIC value.

In the model selection problem of vertex clustering via adjacency spectral embedding, the only two factors that concern us are the dimension of the la-

tent position d and the number of clusters K. So we now consider d and K as the model parameters, which describe the structure of the mixture model. Let $f(\cdot;\theta(d,K))$ be the probability density function of the model which characterizes the rows of extended adjacency spectral embedding. We assume the two models differ from each other if and only if they have distinct model parameters, so selecting a model from the family is equivalent to determining the pair of model parameters. Now we can recast the model selection problem in the simultaneous model selection framework as follows: assume we have a family of D-dimensional distributional models, each with a distinct pair of model parameters (d,K) that determine the structure of the model. Here D is the dimension in the extended adjacency spectral embedding $\hat{Z}$ (defined in (3.11)). The model selection problem is to choose a model by comparing the BIC values BIC$(\hat{Z};d,K)$ evaluated on the observed $\hat{Z}$ throughout all (d,K) pairs. Here BIC$(\hat{Z};d,K)$ is defined by (4.7) with model $f(\cdot;\theta(d,K))$. The model with the largest BIC value is chosen. As we will see in section 4.2.2, the model 1 defined in section 3.4 is a good candidate to characterize the distribution of extended ASE. A summary of simultaneous model selection is presented in algorithm 6.

4.2.2 Consistency of model parameter estimates

In the framework of simultaneous model selection (SMS), a probability model $f(\cdot;\theta(d,K))$ for the rows of extended adjacency spectral embedding is needed.

Algorithm 6 Framework of simultaneous model selection (SMS)

Input: The adjacency matrix $A \in \mathbb{R}^{n \times n}$; an upper bound D of the dimension of latent position; an upper bound $K_{\max}$ of number of clusters

procedure SMS($A, D, K_{\max}$)
  Perform spectral decomposition by $A = \hat{U}_{[D]} \hat{\Lambda}_{[D]} \hat{U}_{[D]}^T$
  Compute $\hat{Z} = \hat{U}_{[\hat{D}]} \hat{\Lambda}_{[\hat{D}]}^{\frac{1}{2}}$
  for $d \leftarrow 1 : D$ **do**
    for $k \leftarrow 1 : K_{\max}$ **do**
      Compute BIC$(\hat{Z}; d, K)$
    end for
  end for
  $(\hat{d}, \hat{K}) \leftarrow \arg\max_{d,K}$ BIC$(\hat{Z}; d, K)$
end procedure

Output: A model $f(\cdot; \theta(\hat{d}, \hat{K}))$ with model parameter $(\hat{d}, \hat{K})$.

The model parameter d should play a similar role as the embedding dimension, which separates the informative dimension and redundant dimension in extended ASE. The model parameter K should be the number of mixture components in the model. If we have such a family of models that well approximates the distribution of the extended ASE with an appropriate (d, K), we can apply SMS procedure described in algorithm 6. Fortunately, model 1 defined in section 3.4 exactly satisfies these requirements for vertex clustering

task via the extended ASE. To see this, let $G \sim \text{SBM}(n, B, \boldsymbol{\pi})$ be the random graph and $\hat{Z} \in \mathbb{R}^{n \times D}$ be the corresponding extended ASE. Let $d_0 = \text{rank}(B)$ be the dimension of latent position, and K_0 given by the dimension of B be the number of blocks in the SBM. In the model $f(\cdot; \theta(d, K))$ described by (5.5)-(5.8), d is the model parameter which decides the size of the first diagonal block in covariance matrix and K is the model parameter which decides the number of components. Most importantly, by theorem 1 and the conjecture in (3.19) based on our simulation, the rows of $\hat{Z}$ approximately follow the distribution $f(\cdot; \theta(d_0, K_0))$. Therefore if we use this family of models in the SMS procedure, we expect the BIC value will be maximized with model parameter (d_0, K_0). In fact, if we assume that the rows of $\hat{Z}$ do asymptotically follow the distribution in the model, we can prove the consistency of the model parameters estimate with our SMS procedure.

Before we state our theorem, we first define some notations. Let

$$f(\cdot; \theta(d, K)) = \sum_{k=1}^{K} \pi^{(k)} \varphi(\cdot; \mu^{(k)}, \Sigma^{(k)}) \tag{4.16}$$

be a family of GMM density functions for a D dimensional random vector, as defined in (5.5)-(5.8), where (d, K) are the model parameters which determine a specific density function. For given constants d_0 and K_0, let $\theta^*(d_0, K_0)$ be a set

of given parameters in the density function (4.16). We define

$$\theta^*(d,K) = \arg \min_{\theta(d,K)\in\Theta(d,K)} D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta(d,K))] \tag{4.17}$$

for all d, K. Here, $D_{\mathrm{KL}}[g||h]$ is the Kullback-Leibler divergence of density h from density g, defined as

$$D_{\mathrm{KL}}[g||h] = \mathbb{E}_{g(\cdot)}\left[\log\left(\frac{g(X)}{h(X)}\right)\right] = \int \log\left(\frac{g(x)}{h(x)}\right) g(x)\mathrm{d}x \tag{4.18}$$

Notice that this definition is self-consistent on $\theta^*(d_0, K_0)$, because $D_{\mathrm{KL}}[g||h] \geq 0$ and equality holds if and only if $g = h$ almost everywhere, by the properties of KL divergence. We say the model (4.16) is *identifiable* on the density $f(\cdot;\theta^*(d_0,K_0))$, if for all $(d,K) \neq (d_0,K_0)$, $f(\cdot;\theta^*(d,K)) \neq f(\cdot;\theta^*(d_0,K_0))$. In other words, there are no identical density functions from the model with different (d,K). Let $\mathrm{BIC}(\hat{Z};d,K)$ denote the BIC evaluated on $\hat{Z}$ with model $f(\cdot;\theta(d,K))$, i.e.

$$\mathrm{BIC}(\hat{Z};d,K) = 2\sum_{i=1}^{n} \log[f(\hat{Z}_i;\hat{\theta}(\hat{Z};d,K))] - \eta(d,K)\log(n) \tag{4.19}$$

where $\eta(d,K)$ is the number of parameters in the model, n is the number of observations in $\hat{Z}$, and $\hat{\theta}(\hat{Z};d,K)$ is the maximum likelihood estimator (MLE)

of the parameters by optimizing the loglikelihood

$$\hat{\theta}(\hat{Z}; d, K) = \arg \max_{\theta(d,K) \in \Theta(d,K)} \frac{1}{n} \sum_{i=1}^{n} \log f(\hat{Z}_i; \theta(d, K)) \tag{4.20}$$

where $\Theta(d, K)$ is the parameter space of the model with given (d, K).

Using the notation we defined above, we here state our theoretical result as follows:

Theorem 2 (Consistency of model parameter estimates)**.** *Let* $\{\hat{Z}^{(n)}\}_{n=1}^{\infty}$ *be a sequence of random matrices, where each element* $\hat{Z}^{(n)} \in \mathbb{R}^{n \times D}$ *is a matrix with* n *rows of* D*-dimensional random vectors. If*

a) Every row in $\hat{Z}^{(n)}$ *is independently identically distributed according to (4.16), with parameter* $\theta^*(d_0, K_0)$*, i.e. for an arbitrary* n*,*

$$\hat{Z}_i^{(n)} \sim f(\cdot; \theta^*(d_0, K_0)) \tag{4.21}$$

i.i.d for all $i \in [n]$.

b) The model $f(\cdot; \theta(d, K))$ *is identifiable on density* $f(\cdot; \theta^*(d_0, K_0))$.

c) For all (d, K)*, the parameter space* $\Theta(d, K)$ *is a compact metric space.*

Then the estimates of model parameters given by

$$(\hat{d}^{(n)}, \hat{K}^{(n)}) = \arg \max_{d \in [D], K \in [K_{max}]} \textit{BIC}(\hat{Z}^{(n)}; d, K) \tag{4.22}$$

(with a constant $K_{max} \geq K_0$) will converge to the truth, i.e.

$$(\hat{d}^{(n)}, \hat{K}^{(n)}) \xrightarrow{p} (d_0, K_0) \tag{4.23}$$

as $n \to \infty$.

To prove this theorem, we begin with the following lemma:

Lemma 1. *Follow the notation in theorem 2, for all* d, K*,*

$$\frac{1}{n}\sum_{i=1}^{n} \log\left[\frac{f(\hat{Z}_i^{(n)}; \theta^*(d_0, K_0))}{f(\hat{Z}_i^{(n)}; \hat{\theta}(d, K))}\right] \xrightarrow{p} D_{KL}[f(\cdot; \theta^*(d_0, K_0))||f(\cdot; \theta^*(d, K))] \tag{4.24}$$

as $n \to \infty$.

Proof. By the definition of Kullback-Leibler divergence in (4.18),

$$\begin{aligned} & D_{\mathrm{KL}}\left[f(\cdot; \theta^*(d_0, K_0))||f(\cdot; \theta^*(d, K))\right] \\ =& \mathbb{E}\left[\log\left(\frac{f(\hat{Z}_i^{(n)}; \theta^*(d_0, K_0))}{f(\hat{Z}_i^{(n)}; \theta^*(d, K))}\right)\right] \\ =& \mathbb{E}\left[\log(f(\hat{Z}_i^{(n)}; \theta^*(d_0, K_0)))\right] - \mathbb{E}\left[\log(f(\hat{Z}_i^{(n)}; \theta^*(d, K)))\right] \end{aligned} \tag{4.25}$$

So we can prove the lemma by showing

$$\frac{1}{n}\sum_{i=1}^{n} \log\left[f(\hat{Z}_i^{(n)}; \theta^*(d_0, K_0))\right] \xrightarrow{p} \mathbb{E}[\log(f(\hat{X}_i; \theta^*(d_0, K_0)))] \tag{4.26}$$

and

$$\frac{1}{n}\sum_{i=1}^{n}\log\left[f(\hat{Z}_i^{(n)};\hat{\theta}(d,K))\right]\xrightarrow{p}\mathbb{E}[\log(f(\hat{X}_i;\theta^*(d,K)))] \tag{4.27}$$

as $n\to\infty$. (4.26) is the direct result of the law of large numbers. (4.27) is the result of theorem 2.2 in [110] then followed by Slutsky's theorem. ■

Now we show the proof of theorem 2 as follows:

Proof of theorem 2. Since $\hat{d}^{(n)}$ and $\hat{K}^{(n)}$ are both integer random variables, to show $(\hat{d}^{(n)},\hat{K}^{(n)})\xrightarrow{p}(d_0,K_0)$ is equivalent to showing

$$\mathbb{P}\left[(\hat{d}^{(n)},\hat{K}^{(n)})=(d_0,K_0)\right]\longrightarrow 1 \tag{4.28}$$

By the definition of $\hat{d}^{(n)}$ and $\hat{K}^{(n)}$ in (4.22), the event $\left\{(\hat{d}^{(n)},\hat{K}^{(n)})=(d_0,K_0)\right\}$ is equivalent to the event $\left\{(d_0,K_0)=\arg\max_{d\in[D],K\in[K_{\max}]}\mathrm{BIC}(\hat{Z}^{(n)};d,K)\right\}$, which is equivalent to $\bigcap_{d,K}\left\{\mathrm{BIC}(\hat{Z}^{(n)};d_0,K_0)\geq\mathrm{BIC}(\hat{Z}^{(n)};d,K)\right\}$, so

$$\begin{aligned}
&\mathbb{P}\left[(\hat{d}^{(n)},\hat{K}^{(n)})=(d_0,K_0)\right]\\
=&\mathbb{P}\left[\bigcap_{d\in[D],K\in[K_{\max}]}\left\{\mathrm{BIC}(\hat{Z}^{(n)};d_0,K_0)\geq\mathrm{BIC}(\hat{Z}^{(n)};d,K)\right\}\right]\\
=&1-\mathbb{P}\left[\bigcup_{d\in[D],K\in[K_{\max}]}\left\{\mathrm{BIC}(\hat{Z}^{(n)};d_0,K_0)<\mathrm{BIC}(\hat{Z}^{(n)};d,K)\right\}\right]\\
\geq&1-\sum_{d\in[D],K\in[K_{\max}]}\mathbb{P}\left[\mathrm{BIC}(\hat{Z}^{(n)};d_0,K_0)<\mathrm{BIC}(\hat{Z}^{(n)};d,K)\right]
\end{aligned} \tag{4.29}$$

Thus in order to show (4.28), it is sufficient to show

$$\mathbb{P}\left[\mathbf{BIC}(\hat{Z}^{(n)}; d_0, K_0) < \mathbf{BIC}(\hat{Z}^{(n)}; d, K)\right] \longrightarrow 0 \tag{4.30}$$

as $n \to \infty$ for all $(d, K) \neq (d_0, K_0)$. By the notation in (4.19) and (4.20), we notice

$$\begin{aligned}
&\frac{1}{2n}\left[\mathbf{BIC}(\hat{Z}^{(n)}; d_0, K_0) - \mathbf{BIC}(\hat{Z}^{(n)}; d, K)\right] \\
=&\frac{1}{2n}\left(2\sum_{i=1}^{n}\log\left[f(\hat{Z}_i^{(n)}; \hat{\theta}(d_0, K_0))\right] - \eta(d_0, K_0)\log(n)\right) \\
&-\frac{1}{2n}\left(2\sum_{i=1}^{n}\log[f(\hat{Z}_i^{(n)}; \hat{\theta}(d, K))] - \eta(d, K)\log(n)\right)
\end{aligned}$$

$$\begin{aligned}
=&\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)}; \hat{\theta}(d_0, K_0))}{f(\hat{Z}_i^{(n)}; \theta^*(d_0, K_0))}\right] - \frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)}; \hat{\theta}(d, K))}{f(\hat{Z}_i^{(n)}; \theta^*(d_0, K_0))}\right] \\
&+\frac{1}{2n}[\eta(d, K) - \eta(d_0, K_0)]\log(n) \\
=&-\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)}; \theta^*(d_0, K_0))}{f(\hat{Z}_i^{(n)}; \hat{\theta}(d_0, K_0))}\right] \\
&+\left(\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)}; \theta^*(d_0, K_0))}{f(\hat{Z}_i^{(n)}; \hat{\theta}(d, K))}\right] - D_{\mathrm{KL}}[f(\cdot; \theta^*(d_0, K_0))||f(\cdot; \theta^*(d, K))]\right) \\
&+\left(D_{\mathrm{KL}}[f(\cdot; \theta^*(d_0, K_0))||f(\cdot; \theta^*(d, K))] + \frac{1}{2n}[\eta(d, K) - \eta(d_0, K_0)]\log(n)\right) \\
=&S_1 + S_2 + S_3
\end{aligned} \tag{4.31}$$

where we let

$$S_1 = -\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)};\theta^*(d_0,K_0))}{f(\hat{Z}_i^{(n)};\hat{\theta}(d_0,K_0))}\right] \tag{4.32}$$

$$S_2 = \frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)};\theta^*(d_0,K_0))}{f(\hat{Z}_i^{(n)};\hat{\theta}(d,K))}\right] - D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta^*(d,K))] \tag{4.33}$$

$$S_3 = D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta^*(d,K))] + \frac{1}{2n}[\eta(d,K)-\eta(d_0,K_0)]\log(n) \tag{4.34}$$

So for any $\epsilon > 0$, by (4.31),

$$\begin{aligned}
&\mathbb{P}\left[\mathbf{BIC}(\hat{Z}^{(n)};d_0,K_0) < \mathbf{BIC}(\hat{Z}^{(n)};d,K)\right] \\
=&\mathbb{P}\left[\frac{1}{2n}\left[\mathbf{BIC}(\hat{Z}^{(n)};d_0,K_0) - \mathbf{BIC}(\hat{Z}^{(n)};d,K)\right] < 0\right] \\
=&\mathbb{P}\left[S_1+S_2+S_3<0\right] \\
\leq&\mathbb{P}[S_1<-\epsilon]+\mathbb{P}[S_2<-\epsilon]+\mathbb{P}[S_3<2\epsilon]
\end{aligned} \tag{4.35}$$

Here, we use the fact that

$$\{S_1+S_2+S_3<0\} \subset \left\{\{S_1<-\epsilon\}\bigcup\{S_2<-\epsilon\}\bigcup\{S_3<2\epsilon\}\right\} \tag{4.36}$$

thus

$$\mathbb{P}\left[S_1+S_2+S_3<0\right] \leq \mathbb{P}[S_1<-\epsilon]+\mathbb{P}[S_2<-\epsilon]+\mathbb{P}[S_3<2\epsilon] \tag{4.37}$$

Now in order to show (4.30), by (4.35), it suffices to show

$$\mathbb{P}[S_1 < -\epsilon] \longrightarrow 0 \tag{4.38}$$

$$\mathbb{P}[S_2 < -\epsilon] \longrightarrow 0 \tag{4.39}$$

$$\mathbb{P}[S_3 < 2\epsilon] \longrightarrow 0 \tag{4.40}$$

For (4.32), by lemma 1,

$$\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)};\theta^*(d_0,K_0))}{f(\hat{Z}_i^{(n)};\hat{\theta}(d_0,K_0))}\right] \xrightarrow{p} D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta^*(d_0,K_0))] = 0 \tag{4.41}$$

So

$$\mathbb{P}\left[-\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)};\theta^*(d_0,K_0))}{f(\hat{Z}_i^{(n)};\hat{\theta}(d_0,K_0))}\right] < -\epsilon\right] \longrightarrow 0 \tag{4.42}$$

For (4.33), also by lemma 1,

$$\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)};\theta^*(d_0,K_0))}{f(\hat{Z}_i^{(n)};\hat{\theta}(d,K))}\right] \xrightarrow{p} D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta^*(d,K))] \tag{4.43}$$

So

$$\mathbb{P}\left[\frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{f(\hat{Z}_i^{(n)};\theta^*(d_0,K_0))}{f(\hat{Z}_i^{(n)};\hat{\theta}(d,K))}\right] - D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta^*(d,K))] < -\epsilon\right] \longrightarrow 0 \tag{4.44}$$

For (4.34), if $(d,K) \neq (d_0,K_0)$, then by the identifiability assumption (b), we

know

$$D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta^*(d,K))] > 0 \tag{4.45}$$

So if we take $\epsilon = \frac{1}{3}D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta^*(d,K))]$, then we have

$$\mathbb{P}\left[D_{\mathrm{KL}}[f(\cdot;\theta^*(d_0,K_0))||f(\cdot;\theta^*(d,K))] + \frac{1}{2n}[\eta(d,K)-\eta(d_0,K_0)]\log(n) < 2\epsilon\right] \longrightarrow 0 \tag{4.46}$$

because $\frac{\log(n)}{2n} \longrightarrow 0$ as $n \longrightarrow \infty$. Combining (4.42), (4.44) and (4.46), we have

$$\mathbb{P}\left[\frac{1}{2n}[\mathrm{BIC}(\hat{Z}^{(n)};d_0,K_0) - \mathrm{BIC}(\hat{Z}^{(n)};d,K)] < 0\right] \longrightarrow 0 \tag{4.47}$$

as $n \longrightarrow \infty$ for all $d \in [D]$ and $K \in K_{\max}$. So we have shown (4.30), which finishes the proof of

$$(\hat{d}^{(n)},\hat{K}^{(n)}) \xrightarrow{p} (d_0,K_0) \tag{4.48}$$

as $n \longrightarrow \infty$. ■

4.3 Vertex clustering via simultaneous model selection

Theorem 2 claims that the estimates of the simultaneous model selection (SMS) procedure are consistent with the truth, if we can construct a model

that adequately interprets the distribution of the data. This strong theoretical support in addition to the advantages of SMS that we have discussed in section 4.2.1 motivates us to conduct vertex clustering via SMS. In this section, we will introduce two vertex clustering algorithms which follow the SMS framework.

4.3.1 SMS based clustering algorithm

Now we seek a method for clustering vertices while utilizing the advantage of simultaneous model selection to complete the entire inference task. Again we focus on dealing with the extended adjacency spectral embedding as the data points to be clustered in the Euclidean space. An accurate model to describe the distribution of extended ASE is crucial in two aspects. First, the conditions in theorem 2 need to be satisfied in order to theoretically ensure the consistency of the model parameter estimates in the model selection procedure. Second, clustering results produced by the model-based clustering technique are reasonable only if the data is well approximated in the clustering procedure. While the theoretical results of the distribution of extended ASE are currently unproven, the model 1 that we have proposed in section 3.4 is very close to the distributional behavior for large graphs by simulation. So in order to perform vertex clustering via SMS, using model 1 to describe the extended ASE is the most favorable option thus far to the best of our knowledge.

This motivates us to present a model-based clustering algorithm via simul-

taneous model selection with a Gaussian mixture model. We call our method MCG. The entire procedure of an MCG algorithm consists of three phases. First, we compute the maximum likelihood estimators of the GMM for each pair of (d, K). The MLEs are used to complete the density function while evaluating the likelihood of the data. We call this phase "parameter fitting". Second, we compute the BIC values for all (d, K) pairs, then choose the one with the largest BIC as the model parameter given the data. We call this phase "model selection". Finally, the likelihoods of all the data points are evaluated within the selected model with fitted parameters. Labels are assigned to each point by the maximum a posterior rule. This phase is called "clustering". We explain the three phases in detail below:

Phase 1: Parameter fitting. Assume the data we observed is the adjacency matrix $A \in \mathbb{R}^{n \times n}$ of a graph. As usual, we first apply the spectral decomposition of A up to D dimension, where D is a preset constant which is supposed to be a loose upper bound of the dimension of latent positions. Let $\hat{Z} \in \mathbb{R}^{n \times D}$ be the extended adjacency spectral embedding. We use model 1 as the family of models from which model selection proceeds. Let K_{max} be another preset constant which is supposed to be a loose upper bound of the number of blocks in the underlying stochastic block model. For each (d, K) pair with $d \in [D]$ and $K \in [K_{\text{max}}]$, we compute the MLEs for the model on the extended ASE. We use the notations defined in model 1, where the density function is denoted

by $f(\cdot;\theta(d,K)) = \sum_{k=1}^{K} \pi^{(k)} \varphi(\cdot;\mu^{(k)},\Sigma^{(k)})$. As a standard approach to computing the MLEs in finite mixture models, the expectation maximization (EM) algorithm [23, 70] is applied here. Specifically, let $\hat{\theta}_{<t>}(d,K) = \{\hat{\pi}^{(k)}_{<t>}, \hat{\mu}^{(k)}_{<t>}, \hat{\Sigma}^{(k)}_{<t>}\}_{k=1}^{K}$ denote the tth iteration of the MLEs in the EM algorithm. We first initialize the estimators (by some way that will be discussed in section 4.3.2) with $\hat{\theta}_{<0>}(d,K)$ = $\{\hat{\pi}^{(k)}_{<0>}, \hat{\mu}^{(k)}_{<0>}, \hat{\Sigma}^{(k)}_{<0>}\}_{k=1}^{K}$. Then we alternately apply the E step and M step. In the E step, conditional distribution of the membership $\hat{\tau}_i$ given observation $\hat{Z}_i$ is calculated by

$$T_{i,k}^{<t+1>} = \mathbb{P}[\hat{\tau}_i = k|\hat{Z}_i] = \frac{\hat{\pi}^{(k)}_{<t>}\varphi(\hat{Z}_i;\hat{\mu}^{(k)}_{<t>},\hat{\Sigma}^{(k)}_{<t>})}{\sum_{j=1}^{K}\hat{\pi}^{(j)}_{<t>}\varphi(\hat{Z}_i;\hat{\mu}^{(j)}_{<t>},\hat{\Sigma}^{(j)}_{<t>})} \tag{4.49}$$

In the M step, we calculate the MLE depending on $T_{i,k}$ by

$$\hat{\pi}^{(k)}_{<t+1>} = \frac{1}{n}\sum_{i=1}^{n} T_{i,k}^{<t+1>} \tag{4.50}$$

$$\hat{\mu}^{(k)}_{<t+1>}[1:d] = \frac{\sum_{i=1}^{n} T_{i,k}^{<t+1>}\hat{Z}_i[1:d]}{\sum_{i=1}^{n} T_{i,k}^{<t+1>}} \tag{4.51}$$

$$\hat{\mu}^{(k)}_{<t+1>}[d+1:D] = 0 \tag{4.52}$$

$$\hat{\hat{\Sigma}}^{(k)}_{<t+1>} = \frac{\sum_{i=1}^{n} T_{i,k}^{<t+1>}(\hat{Z}_i[1:d] - \hat{\mu}^{(k)}_{<t+1>}[1:d])(\hat{Z}_i[1:d] - \hat{\mu}^{(k)}_{<t+1>}[1:d])^T}{\sum_{i=1}^{n} T_{i,k}^{<t+1>}} \tag{4.53}$$

$$\hat{\sigma}^{2^{(k)}}_{<t+1>} = \frac{\sum_{i=1}^{n} T_{i,k}^{<t+1>}\hat{Z}_i[d+1:D]^T\hat{Z}_i[d+1:D]}{(D-d)\sum_{i=1}^{n} T_{i,k}^{<t+1>}} \tag{4.54}$$

where $\xi[a:b]$ denotes the a truncated vector of ξ from the a-th entry to the b-th entry. In the end of each iteration of the loop, we terminate the EM-steps

if the difference of the joint log-likelihood of all n data points between the current and the last iteration is small enough, or if the number of iterations reaches a threshold. After the iterations terminate, we take the final estimators $\hat{\theta}(\hat{Z}; d, K) = \{\hat{\pi}^{(k)}, \hat{\mu}^{(k)}, \hat{\Sigma}^{(k)}\}_{k=1}^{K}$ as the MLEs of the model $f(\cdot; \theta(d, K))$. This finishes the parameter fitting phase.

Phase 2: Model selection. After phase 1 we have the MLEs of the model $f(\cdot; \theta(d, K))$ for all (d, K) pairs, thus we can compute $\text{BIC}(\hat{Z}; d, K)$ by equation (4.19). As stated in the general simultaneous model selection procedure in algorithm 6, model parameters (d, K) is estimated by maximizing the BIC values, i.e.

$$(\hat{d}, \hat{K}) \leftarrow \arg\max_{d,K} \text{BIC}(\hat{Z}; d, K) \tag{4.55}$$

In practice, the BIC values may be perturbed by noise or by the instability of the EM algorithm. Sometimes the global maximum of the BIC may arise as an outlier. In these cases we may use regression techniques in lieu of enumerating all BICs to find the local maximizer of the BICs, which we will discuss in section 5.1.2. The estimation of $(\hat{d}, \hat{K})$ determines the model out of the whole family, which finishes the model selection phase.

Phase 3: Clustering. According to the previous results, the model is selected in phase 2 and its MLEs is computed in phase 1. By applying the plug-in rule, we have a specific model with the MLEs as its parameter, namely

$f\left(\cdot;\hat{\theta}(\hat{Z};\hat{d},\hat{K})\right)$. Notice that all of the parameters in this Gaussian mixture model have been assigned values. By the framework of model-based clustering, each row of $\hat{Z}$ can be clustered by the maximum a posteriori rule, i.e.

$$\hat{\tau}_i = \arg\max_j \left\{ \frac{\hat{\pi}_j \varphi(\hat{Z}_i;\hat{\mu}_j,\hat{\Sigma}_j)}{\sum_{k=1}^{\hat{K}} \hat{\pi}_k \varphi(\hat{Z}_i;\hat{\mu}_k,\hat{\Sigma}_k)} \right\} \tag{4.56}$$

This finalizes the inference task of vertex clustering.

A summary of the algorithm MCG is shown in algorithm 7.

Algorithm 7 Model-based clustering algorithm via SMS with GMM (MCG)

Input: The adjacency matrix $A \in \mathbb{R}^{n\times n}$; an upper bound D of embedding dimension; an upper bound $K_{\max}$ of mixture complexity

1: **function** MCG($A, D, K_{\max}$)
2: Apply extended ASE on A with dimension D: $\hat{Z} \leftarrow \hat{U}_{[D]}\hat{\Lambda}_{[D]}^{\frac{1}{2}}$
3: **loop**
4: Compute MLEs $\hat{\theta}(\hat{Z};d,K) = \{\hat{\pi}^{(k)},\hat{\mu}^{(k)},\hat{\Sigma}^{(k)}\}_{k=1}^{K}$ for model 1
5: **end loop**
6: $(\hat{d},\hat{K}) \leftarrow \arg\max_{d\in[D],K\in[K_{\max}]} \text{BIC}(\hat{Z};d,K)$
7: $\hat{\tau}_i = \arg\max_j \left\{ \frac{\hat{\pi}_j \varphi(\hat{Z}_i;\hat{\mu}_j,\hat{\Sigma}_j)}{\sum_{k=1}^{\hat{K}} \hat{\pi}_k \varphi(\hat{Z}_i;\hat{\mu}_k,\hat{\Sigma}_k)} \right\}$
8: **end function**

Output: The clustering label $(\hat{\tau}_1,\ldots,\hat{\tau}_n)$.

Here we present our second model-based clustering algorithm via SMS frame-

work. The MCG algorithm uses both the informative and redundant dimensions of the extended ASE in both the model selection and clustering procedures. Although we believe simultaneous model selection has advantages compared to consecutive model selection, it is unclear whether including redundant dimensions of extended ASE in the clustering procedure is favorable or not. The reason can be explained by two aspects. First, the redundant dimensions may contain little information for the clustering, so including the redundant dimensions might degrade the results of clustering. Second, choosing a smaller dimension in a clustering task may lead to better performance, especially for small number of observations, due to bias-variance tradeoff [44]. This motivates the variation of the third phase in the MCG algorithm.

To be specific, in phase 1 and phase 2, model 1 and extended ASE $\hat{Z}$ are utilized just to find the estimate of the embedding dimension. In phase 3, we can now truncate the extended ASE by the dimension $\hat{d}$ which is estimated by the SMS procedure. In this context, redundant dimensions do not take part in the clustering procedure. It follows that we may apply the model-based clustering algorithm, summarized in algorithm 4, on the truncated embedding $\hat{Z}_{\hat{d}}$ with a regular Gaussian mixture model. Notice that the embedding dimension is determined by the SMS procedure, so the clustering results could be remarkably different than the algorithm under the consecutive model selection framework. We call this algorithm MCEG, inspired by model-based clustering by GMM

with an embedding dimension determined via SMS. An outline detailing the steps of MCEG is shown in algorithm 8.

Algorithm 8 Model-based clustering by GMM with embedding dimension determined via SMS (MCEG)

Input: The adjacency matrix $A \in \mathbb{R}^{n\times n}$; an upper bound D of embedding dimension; an upper bound $K_{\max}$ of mixture complexity

function MCG($A, D, K_{\max}$)
  Apply extended ASE on A with dimension D: $\hat{Z} \leftarrow \hat{U}_{[D]}\hat{\Lambda}_{[D]}^{\frac{1}{2}}$
  loop
    Compute MLEs $\hat{\theta}(\hat{Z}; d, K) = \{\hat{\pi}^{(k)}, \hat{\mu}^{(k)}, \hat{\Sigma}^{(k)}\}_{k=1}^{K}$ for model 1
  end loop
  $\hat{d} \leftarrow \arg\max_{d\in[D],K\in[K_{\max}]} \text{BIC}(\hat{Z}; d, K)$
  Truncate the ASE: $\hat{Z}_{[\hat{d}]} \leftarrow \hat{U}_{[\hat{d}]}\hat{\Lambda}_{[\hat{d}]}^{\frac{1}{2}}$
  $\hat{K} \leftarrow \text{BIC}(\hat{Z}_{[\hat{d}]})$ by algorithm 3
  $(\hat{\tau}_1, \ldots, \hat{\tau}_n) \leftarrow \text{GMM}(\hat{Z}_{[\hat{d}]}, \hat{K})$ by algorithm 4
end function

Output: The clustering label $(\hat{\tau}_1, \ldots, \hat{\tau}_n)$.

4.3.2 Initialization and convergence

In phase 1 of MCG and MCEG algorithms presented above, we use the EM algorithm to compute the MLEs. To initialize the EM iterations for a fixed

pair (d, K), we calculate the parameters $\hat{\theta}_{<0>}(d, K) = \{\hat{\pi}^{(k)}_{<0>}, \hat{\mu}^{(k)}_{<0>}, \hat{\Sigma}^{(k)}_{<0>}\}_{k=1}^{K}$ by preliminary clustering results. The clustering results divide the n data points into K groups. Let $\hat{\tau}_i^{<0>}$ be the group assignment label for the i-th data point in the preliminary clustering result. Then let the initial conditional distribution of the membership be

$$T_{i,k}^{<0>} = \begin{cases} 1, & k = \hat{\tau}_i^{<0>} \\ 0, & k \neq \hat{\tau}_i^{<0>} \end{cases} \tag{4.57}$$

Then we can calculate the initial parameters $\hat{\theta}_{<0>}(d, K)$ by (4.50)-(4.54).

Since different preliminary clustering methods may affect the final results of our method, we want to see how robust our method is with respect to initialization. We mainly tried three preliminary clustering methods:

1) **Random**. We randomly divide the n data points into K groups, with each group holding approximately n/K number of points. For this method, we do not use any of the information from the data for the preliminary clustering.

2) **Mclust**. We apply the Mclust method [29], an BIC algorithm implemented in an R package, to the first d dimension of the observed data points with fixed K, then we take the clustering result as the preliminary clustering result. Note that Mclust uses hierarchical clustering as its preliminary initialization.

3) **Kmeans**. We apply the Kmeans algorithm to the first d dimension of the observed data points with fixed K, then we take the clustering result as the preliminary clustering result.

In order to evaluate the robustness of different initialization, we perform a simulation with our method. The random graph is drawn from an SBM with fixed $B = \begin{bmatrix} 0.2 & 0.09 \\ 0.09 & 0.1 \end{bmatrix}$, prior block probability $\pi = (0.5, 0.5)^T$, and $n = 200$ and $n = 500$ fixed. After 100 Monte Carlo trials for each n, we get the following results: For $n = 200$, all three initialization methods give the same $\hat{d} = 1$ and $\hat{K} = 4$ throughout all Monte Carlo trials, with sightly different but very similar ARI. For $n = 500$, all three initialization methods give the same $\hat{d} = 2$ and $\hat{K} = 2$ throughout all Monte Carlo trials with exactly the same ARI. This result shows that our method is robust with respect to initializations. The only concern may involve the speed of the convergence in the EM algorithm. For this reason we did another simulation to see how fast the three initializations converge. Using the same settings in the previous experiment, but we vary the number of vertices n from 200 to 10000. The result shows that for any n, all three initializations will eventually converge to the same loglikelihood. For small n we observe that Random and Mclust converge faster than Kmeans, and for large n Mclust and Kmeans converge faster than Random. Later in the experiment, we use Mclust as the default initialization method of the EM algorithm.

4.4 Numerical results on synthetic data

In this section, we evaluate the performance of MCG and MCEG algorithms (see Algorithm 7 and Algorithm 8) by simulations on synthetic data. We compare our methods with the BIC $\circ$ ZG methods, the combination of an ubiquitous GMM approach via consecutive model based clustering (see Algorithm 5). Since a constant l is required as an input in the ZG algorithm in order to determine which elbow of the scree plot is taken, we refer ZGl and BIC $\circ$ ZGl to the algorithms for given l. The job of deciding the ordinal number of the elbow is always subjective in practice, so we will consider ZG1, ZG2 and ZG3, the ZG algorithm which takes the 1st, 2nd and 3rd elbows respectively, at the same time in competition. Notice that even if one ZGl (or corresponding BIC $\circ$ ZGl) method outperforms our proposed SMS method in a specific setting, it does not mean that the ZG algorithm is superior to ours because the optimal l may be changed in a different setting. We will see this in the simulation. We apply the Mclust R package [29] to perform BIC algorithm. Additionally, we also perform two well-known heuristic vertex clustering methods for comparison. One is the Louvain algorithm proposed in [10]; the other is the Walktrap algorithm proposed in [83].

There are numerous criteria to evaluate the performance of a clustering result, including Jaccard [42], rand index [41], normalized mutual informa-

tion [20] and variation of information [73]. Of these, we choose the well known adjusted rand index (ARI) as the measure of the similarity between the clustering result and the ground truth labels. As a corrected-for-chance version of the rand index, ARI normalizes the rand index so that the expected value of that between a random cluster and the ground truth is zero. The maximum value of ARI is 1, which indicates perfect agreement of two partitions. So a larger ARI means the clustering is performing better.

4.4.1 Simulations on GMM data

In the simulation, we evaluate MCG/MCEG algorithms under the assumption that the extended ASE of an SBM graph exactly follows a Gaussian mixture model, as the conjecture says in (3.24). So the rows of the data matrix $W \in \mathbb{R}^{n \times D}$ are generated i.i.d from a GMM, i.e.

$$W_i \sim f(\cdot; \theta(d, K)) \tag{4.58}$$

where the parameters $\theta(d, K)$ follow the formula in Theorem 1 and the structure of Model 1. That is, the GMM is determined as long as we fix the latent positions $X^{(k)} \in \mathbb{R}^{1 \times d}$ and the prior block probability $\pi^{(k)}$, for $k = 1, \ldots, K$. By Model 1, $\theta(d, K) = \left\{ \pi^{(k)}, \left[\mu_1^{(k)}, \ldots, \mu_d^{(k)} \right], \tilde{\Sigma}^{(k)}, \sigma^{2^{(k)}} \right\}_{k=1}^{K}$. Given $X^{(k)}$ and $\pi^{(k)}$, we know $\left[\mu_1^{(k)}, \ldots, \mu_d^{(k)} \right] = X^{(k)}$, $\tilde{\Sigma}^{(k)}$ is given by (3.13), and $\sigma^{2^{(k)}}$ can be estimated by

simulation (as shown in Observation 4 in Section 3.3.2). Thus we can design a principled data set by discussion on $X^{(k)}$ and $\pi^{(k)}$.

Case 1: Full rank block probability matrix. In this case, we let $d = 2$ and $K = 2$, $\pi^{(1)} = \pi^{(2)} = 0.5$. Let

$$X^{(1)} = [0.5, 0]$$

$$X^{(2)} = [0.3\cos(\theta), 0.3\sin(\theta)]$$

Notice that the block probability matrix $B = XX^T$ is full rank.

First, we fix $n = 200$, and vary $\theta = \pi/6$, $\pi/4$ and $\pi/3$. The mean of ARI for different methods is shown in Table 4.1. We can see that MCG wins over other methods in all cases. The ARI become higher as we increase the angle between the two latent vectors. This is because larger angle results in more distinguishable blocks. The accuracy of the estimation of d for different methods is shown in Table 4.2. Again MCG is the best method, and it has 100% accuracy. If we fix $\theta = \pi/4$ and vary $n = 200, 300, 400, 500$, the results are shown in Table 4.3. Still our methods outperform the others.

Case 2: Low rank block probability matrix. Now we let $d = 2$, $K = 3$,

method\θ	$\pi/6$	$\pi/4$	$\pi/3$
MCG	**0.873**	**0.939**	**0.992**
MCEG	0.832	0.884	0.976
BIC $\circ$ ZG1	0.769	0.897	0.974
BIC $\circ$ ZG2	0.731	0.893	0.984
BIC $\circ$ ZG3	0.156	0.899	0.988

Table 4.1: The mean of ARI for different methods in a full rank case. Data is generated from a GMM with varying latent position angle θ. The number of points is fixed as $n = 200$.

method\θ	$\pi/6$	$\pi/4$	$\pi/3$
MCG	100%	100%	100%
ZG1	0%	0%	0%
ZG2	0%	0%	0%
ZG3	0%	0%	0%

Table 4.2: The accuracy of the estimation of d_0 for different methods in full rank case. Data is generated from a GMM with varying latent position angle θ. The number of points is fixed as $n = 200$.

$\rho^{(1)} = 0.4$, $\rho^{(2)} = 0.3$, $\rho^{(3)} = 0.3$.

$$Z^{(1)} = [0.5, 0]$$

$$Z^{(2)} = [0.3\cos(\pi/4), 0.3\sin(\pi/4)]$$

$$Z^{(3)} = [0.4\cos(\theta), 0.4\sin(\theta)]$$

method$\backslash n$	200	300	400	500
MCG	**0.939**	**0.981**	**0.990**	**0.997**
MCEG	0.884	0.967	0.983	0.993
BIC $\circ$ ZG1	0.897	0.963	0.984	0.992
BIC $\circ$ ZG2	0.893	0.967	0.985	**0.997**
BIC $\circ$ ZG3	0.899	0.972	0.989	**0.997**

Table 4.3: The mean of ARI for different methods in a full rank case. Data is generated from a GMM with fixed latent position angle $\theta = \pi/4$. The number of points vary from 200 to 500.

In this setting, the communication matrix $B = XX^T$ has rank 2, which is not a full rank matrix. For this reason we call this the low rank case. If we fix $n = 200$ and vary $\theta = \pi/6$, $\pi/4$ and $\pi/3$, the result is shown in Table 4.4 and Table 4.5. If we fix $\theta = \pi/4$ and vary $n = 200, 300, 400, 500$, the result is shown in Table 4.6. We find that MCG is again the best one in most cases. Both the full rank and low rank cases demonstrate the superiority of MCG.

method$\backslash\theta$	$\pi/6$	$\pi/4$	$\pi/3$
MCG	**0.460**	**0.447**	**0.563**
MCEG	0.133	0.378	0.541
BIC $\circ$ ZG1	0.249	0.429	0.536
BIC $\circ$ ZG2	0	0.378	0.541
BIC $\circ$ ZG3	0	0	0.542

Table 4.4: The mean of ARI for different methods in a low rank case. Data is generated from a GMM with varying latent position angle θ. The number of points is fixed as $n = 200$.

method\θ	$\pi/6$	$\pi/4$	$\pi/3$
MCG	100%	100%	100%
ZG1	0%	0%	0%
ZG2	0%	100%	100%
ZG3	0%	0%	0%

Table 4.5: The accuracy of the estimation of d_0 for different methods in low rank case. Data is generated from a GMM with varying latent position angle θ. The number of points is fixed as $n = 200$.

method\n	200	300	400	500
MCG	**0.447**	0.506	**0.688**	**0.750**
MCEG	0.378	0.508	0.590	0.610
BIC $\circ$ ZG1	0.429	**0.510**	0.576	0.602
BIC $\circ$ ZG2	0.378	0.445	0.590	0.610
BIC $\circ$ ZG3	0	0.412	0.575	0.614

Table 4.6: The mean of ARI for different methods in a low rank case. Data is generated from a GMM with fixed latent position angle $\theta = \pi/4$. The number of points vary from 200 to 500.

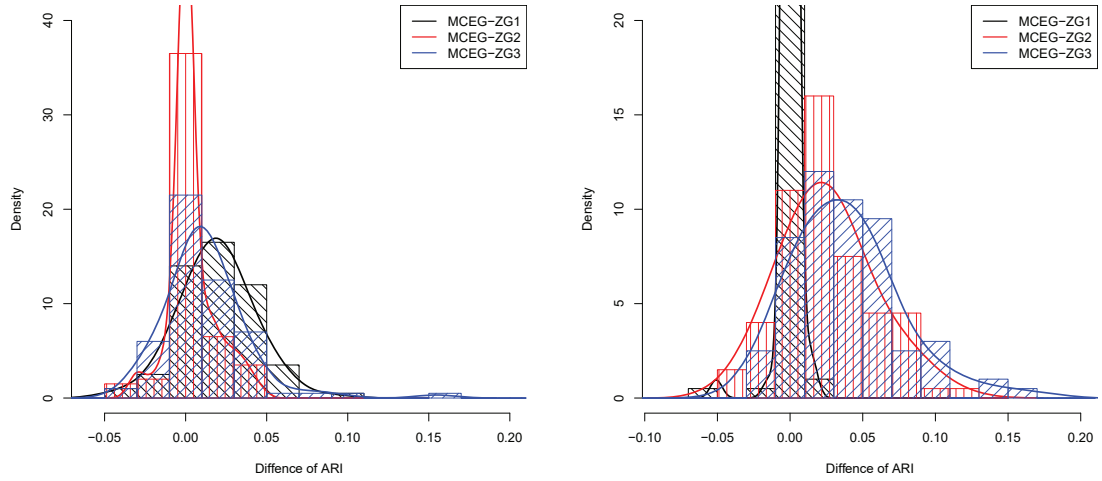
4.4.2 Simulations on SBM data

Now we generate a graph G from a stochastic block model SBM(n, B, π) by specifying the block probability matrix B, prior block probability π and number of vertices n. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ represents G. Then we apply the extended adjacency spectral embedding on the graph, denoted by $\hat{Z} \in \mathbb{R}^{n \times D}$. For simplicity, we fix $D = 8$.

Case 1: Full rank block probability matrix. Let $n = 500$, $B = \begin{bmatrix} 0.2 & p \\ p & 0.1 \end{bmatrix}$, $\pi = (0.5, 0.5)$. We vary p to change the angle between two latent vectors.

Figure 4.1 shows the difference of ARI between MCEG and BIC $\circ$ ZG methods in histograms and density curves. All of the differences are paired for 100 Monte Carlo trials. A point appearing to be bigger than 0 means MCEG has higher ARI than the corresponding BIC $\circ$ ZG method in that Monte Carlo replica. So if the histogram tends to be on the right side of 0, we can tell MCEG is better. Figure 4.1(a) shows the result under the setting with $p = 0.095$. We find MCEG outperforms BIC $\circ$ ZG1 and BIC $\circ$ ZG3, and has many ties with BIC $\circ$ ZG2. We did a sign test for the paired differences of ARI, where the null hypothesis is that the two methods are equally good or BIC $\circ$ ZG is better ($p \leq 0.5$ with respect to Binomial distribution), and the alternative hypothesis is that our method is better ($p > 0.5$). We ignore ties in the sign test. The p-values for MCEG comparing to BIC $\circ$ ZG1, BIC $\circ$ ZG2 and BIC $\circ$ ZG3 are $2.013e-12$, 0.04035 and $2.674e-07$ respectively. The small p-values suggest MCEG is statistically significantly better than those BIC $\circ$ ZG methods, even for BIC $\circ$ ZG2. In figure 4.1(b), $p = 0.115$ in B matrix. Similarly, the p-values of a sign test for MCEG comparing to BIC $\circ$ ZG1, BIC $\circ$ ZG2 and BIC $\circ$ ZG3 are 0.3437, $5.446e-09$ and $2.967e-15$ respectively. In this case, MCEG has similar performance with BIC $\circ$ ZG1, but outperforms BIC $\circ$ ZG2 and BIC $\circ$

ZG3. In both cases, MCEG has the best performance with respect to ARI. In contrast, none of the BIC ∘ ZG methods win in both cases. Considering that in practice we need to fix an elbow in BIC ∘ ZG methods without knowing the ground truth, MCEG is a more robust algorithm.



(a) $p = 0.095$ (b) $p = 0.115$

Figure 4.1: The difference of ARI between MCEG and BIC ∘ ZG methods shown in histograms and density curves. Random graphs are generated from a 2-block SBM with a full rank block probability matrix. The number of vertices is fixed as $n = 500$. The between block probability p varies: (a) $p = 0.095$, (b) $p = 0.115$.

Figure 4.2 shows the mean of ARI for all methods, including the existing heuristic Louvain and Walktrap algorithms. The random graph with $n = 500$ vertices is generated from a 2-block SBM with block probability matrix [0.2, p; p, 0.1]. The parameter p is varying from 0.09 to 0.115. We observe that the Louvain and Walktrap algorithms do not perform well for large p, so we may

conclude that these two heuristic vertex clustering algorithms are not suitable for specific SBM graphs. To have a detailed look, Figure 4.3 shows the mean of ARI for MCEG and ZGs. In figure 4.3(a), all methods have decreasing ARI as p going up. This is because the angle between two latent vectors become smaller, so the clustering centers get closer. Out of all the methods, our proposed MCEG performs well in all p's. In figure 4.3(b), the mean of $\hat{d} - d$ is plotted. We can see that MCEG is the closest one to zero, which means it estimates $\hat{d}$ better than the other methods.

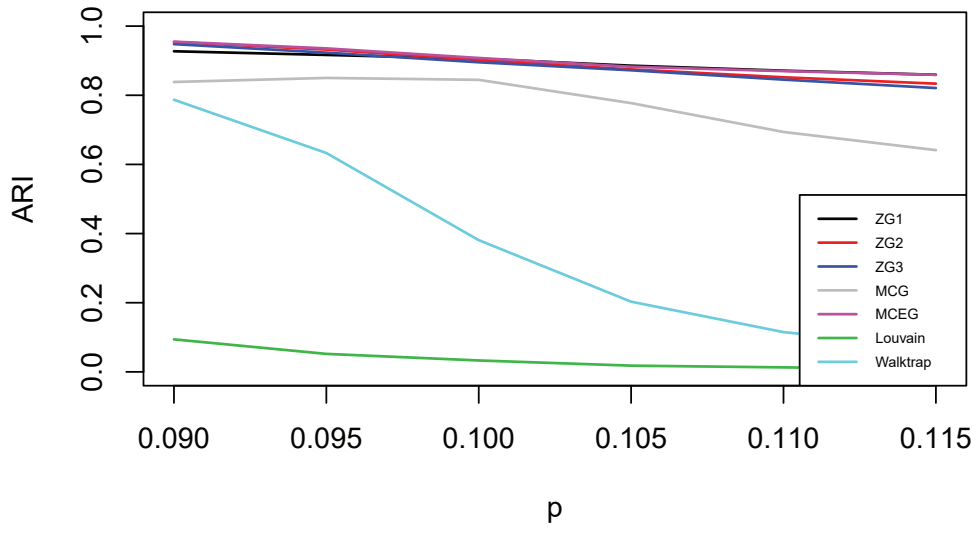
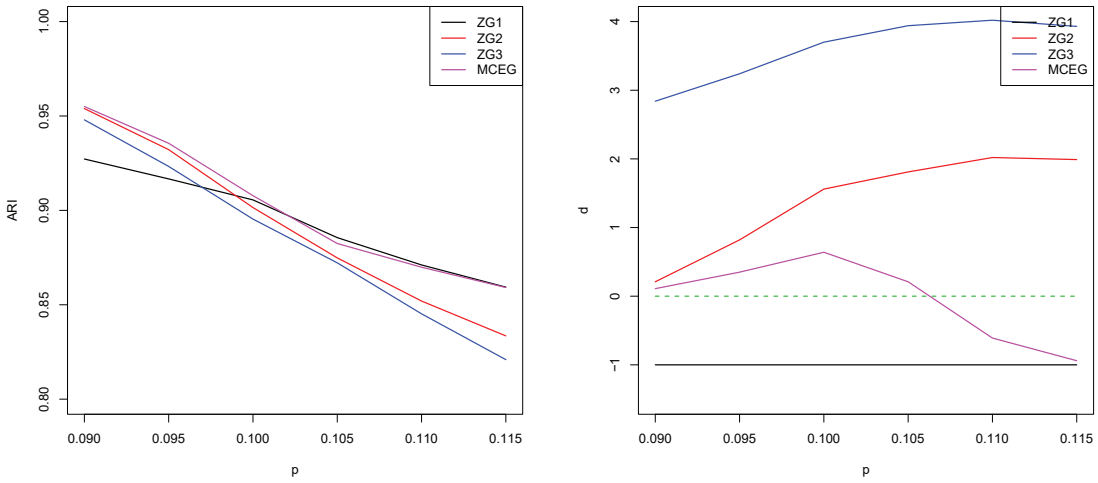


Figure 4.2: The mean of ARI of 100 Monte Carlo trials for different methods. The random graph with $n = 500$ vertices is generated from a 2-block SBM with block probability matrix [0.2, p; p, 0.1]. The parameter p is varying from 0.09 to 0.115.



(a) mean of ARI

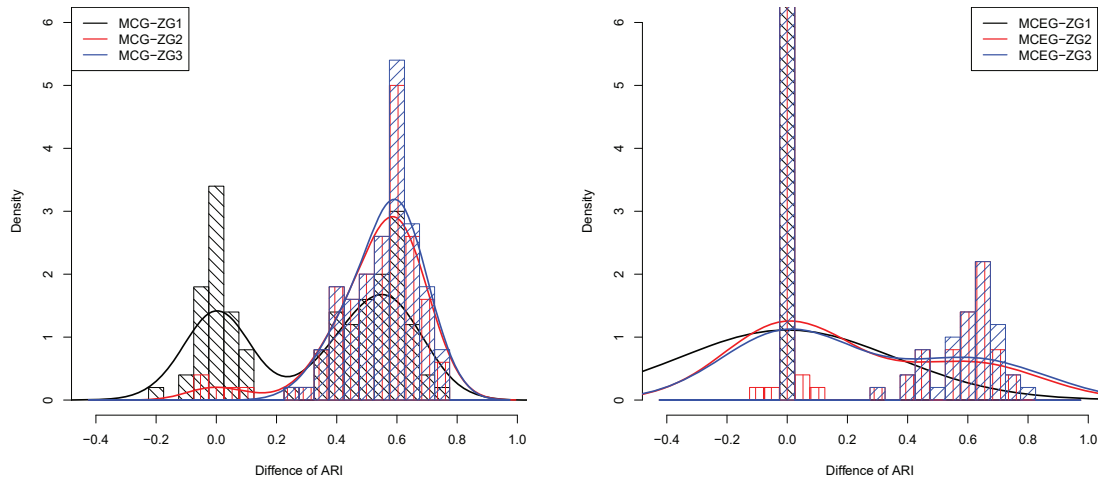
(b) mean of $\hat{d} - d$

Figure 4.3: The mean of ARI and $\hat{d}$ for varying p: (a) mean of ARI (b) mean of $\hat{d} - d$. The random graph with $n = 500$ vertices is generated from a 2-block SBM with block probability matrix [0.2, p; p, 0.1]. The parameter p is varying from 0.09 to 0.115.

Case 2: Low rank block probability matrix. Let $n = 200$, $B = \begin{bmatrix} p^4 & p^3 \\ p^3 & p^2 \end{bmatrix}$ where $p = 0.9$, $\pi = (0.5, 0.5)$. In this case, $d = \text{rank}(B) = \text{rank}(P) = 1$.

Figure 4.4 shows the difference of ARI between our methods and BIC $\circ$ ZG methods in histograms and density curves. Similarly to Figure 4.1, the points being greater than 0 means our method outperforms the other one with respect to ARI. Figure 4.4(a) is the comparison of MCG with BIC $\circ$ ZG methods. We see MCG beats BIC $\circ$ ZG2 and BIC $\circ$ ZG3 in almost all the Monte Carlo replica, and it beats BIC $\circ$ ZG1 in at least half of the trials. The p-values of a signed test for MCG compared to BIC $\circ$ ZG1, BIC $\circ$ ZG2 and BIC $\circ$ ZG3 are $1.436e-09$,

$2.2e-16$ and $2.2e-16$ respectively. Figure 4.4(b) is the comparison of MCEG with BIC $\circ$ ZG methods. The p-values of a sign test for MCEG compared to BIC $\circ$ ZG2 and BIC $\circ$ ZG3 are $7.467e-10$ and $9.095e-13$ respectively. MCEG ties with BIC $\circ$ ZG1 in all the Monte Carlo trials, but it outperforms BIC $\circ$ ZG2 and BIC $\circ$ ZG3.

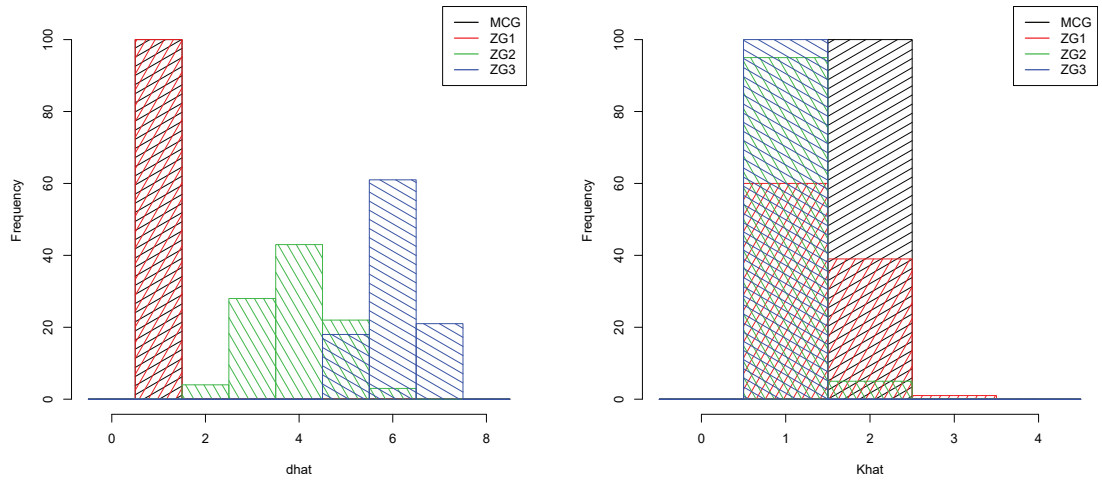


(a) MCG-BIC $\circ$ ZG (b) MCEG-BIC $\circ$ ZG

Figure 4.4: The difference of ARI between MCG/MCEG and BIC $\circ$ ZG methods shown in histograms and density curves. Random graphs are generated from a 2-block SBM with a low rank block probability matrix. The number of vertices is fixed as $n = 200$. (a) MCG-BIC $\circ$ ZG, (b) MCEG-BIC $\circ$ ZG.

Figure 4.5 shows the estimators of embedding dimension $\hat{d}$ and number of clusters $\hat{K}$ for each methods. Since we know the true $d = 1$ and $K = 2$, we can see from these figures the accuracy of the algorithms estimating the model parameters. In Figure 4.5(a), we observe that MCG and ZG1 have 100% accuracy in getting the correct estimation of $d = 1$, while ZG2 and ZG3 always overesti-

mate it. The estimation of K is shown in Figure 4.5(b). MCG again has 100% accuracy of estimation, while the others do not. Since an incorrect estimation of number of clusters will give rise to a severe influence of the clustering result, this is why our methods have much better performance in terms of ARI. Notice that the estimation of d for MCEG is exactly the same as MCG, so we only show MCG in this figure.



(a) $\hat{d}$ (b) $\hat{K}$

Figure 4.5: The estimates of embedding dimension $\hat{d}$ and number of clusters $\hat{K}$ shown in histograms for different method. Random graphs are generated from a 2-block SBM with a low rank block probability matrix. The number of vertices is fixed as $n = 200$. (a) $\hat{d}$, (b) $\hat{K}$

Chapter 5

Demonstration of Vertex Clustering on Connectomics

In this chapter, we demonstrate the performance of MCG and MCEG algorithms via the simultaneous model selection procedure on real data sets of connectomes, a sort of brain graph induced from brain neuronal connections. For the origin of the term *connectomes*, we refer the readers to [35, 97]. Basically, the connectome describes the network of the brain consisting of neurons (or collections thereof) as vertices and synapses (or structural connections) as edges. It is fundamentally helpful to unlock the structural and functional unknowns in the human brain in cognitive neuroscience and neuropsychology by studying the topological properties of the connectome. We apply our simultaneous model selection approach on both macro-scale and micro-scale connectome

data sets to provide statistical analysis of the clustering structure of the brain network. These two types of connectomes represent undirected graphs and directed graphs respectively in the real-world application.

5.1 Human connectomes

5.1.1 Data description

In this section, we study our methods on a data set of human connectomes. The raw data is collected by the diffusion magnetic resonance imaging (dMRI), which can present the structural connectivity within the brain. The macro-scale connectomes are estimated by a NeuroData's MRI to graphs (NDMG) pipeline [50], which is designed to produce robust and biologically plausible connectomes across studies, individules and scans. The raw diffusion MRI data is processed through the pipeline following four steps: registration, tensor estimation, tractography and graph generation. As the output of NDMG pipeline, the brain graphs, namely connectomes, are generated. The vertices of the graph represent regions of interest (ROI) gained by spatial proximity, and the edges of the graph represents the connection between ROIs via tensor-based fiber streamlines. Specifically, there is an edge for a pair of ROIs if and only if there is a streamline passing between them. The graph is undirected

since the raw dMRI data does not have direction information. The connectomes are generated across multiple parcellations, which result in 24 different spatial resolutions. In our demonstration, we pick the graphs in study BNU1 with parcellation DS01216 under the consideration of medium graph size. For more details of the data set, we refer the readers to [50].

This specific data set with parcellation DS01216 consists of 114 connectomes (57 subjects with 2 scans each), with 1215 vertices for each graph. There are two attributes for each vertex, the regions of interest, in the graph. One attribute is hemisphere, which could be either left, right or other. The other attribute is tissue, which could be either gray, white or other. For ease of illustration, we only consider the regions in left or right hemisphere, and in gray or white tissue. So we get an induced subgraph from the original connectome by deleting the vertices labeled "other" in hemisphere or tissue attributes. Then we extract the largest connected component of that subgraph so as to support the adjacency spectral embedding. This yields 114 connected undirected graphs, with approximately 760 vertices for each graph. Each vertex has been assigned two labels, one represents the hemisphere and the other represents the tissue. They are treated as the ground truth of the clustering structure in the graph. We apply our simultaneous model selection methods followed by model-based clustering approach, described in algorithm 7 and algorithm 8, on the 114 graphs.

5.1.2 Maximizing BIC values via regression

In the experiment, we perform simultaneous model selection (SMS) on the 114 graphs following the framework shown in algorithm 6. Model 1 is used in the SMS procedure since the graphs are undirected. There are three inputs for the SMS procedure SMS$(A, D, K_{\max})$, namely the adjacency matrix A, the upper bound of dimension of latent position D, and the upper bound of number of clusters $K_{\max}$. After the preprocessing steps, the adjacency matrix $A \in \mathbb{R}^{n \times n}$ with $n \approx 760$ represents each graph. We set $D = 100$, which means we perform spectral decomposition on A and extended adjacency spectral embedding $\hat{Z}$ with dimension $D = 100$. We set $K_{\max} = 30$, which is much larger than the number of clusters in the ground truth. Also in practice, we introduce another algorithm input $d_{\max}$, which indicates the test range of the model parameter d (see the definition of d in model 1). In other words, we will compute BIC values for each pair of model parameters (d, K) with $d = 1 \ldots, d_{\max}$ and $K = 1 \ldots K_{\max}$. The reason we scan d just up to $d_{\max}$ but not D is the consideration of computational cost.

By the description of SMS, we estimate the model parameters d and K by maximizing the BIC values over all (d, K) pairs, i.e. $(\hat{d}, \hat{K}) = \arg\max_{d,K} \text{BIC}(\hat{Z}; d, K)$. For real data, however, the BIC values may be perturbed by noise, corruption or outliers. In order to mitigate those unexpected effects, we smooth the BIC values by regression. To be specific, we first maximize the BIC values over K

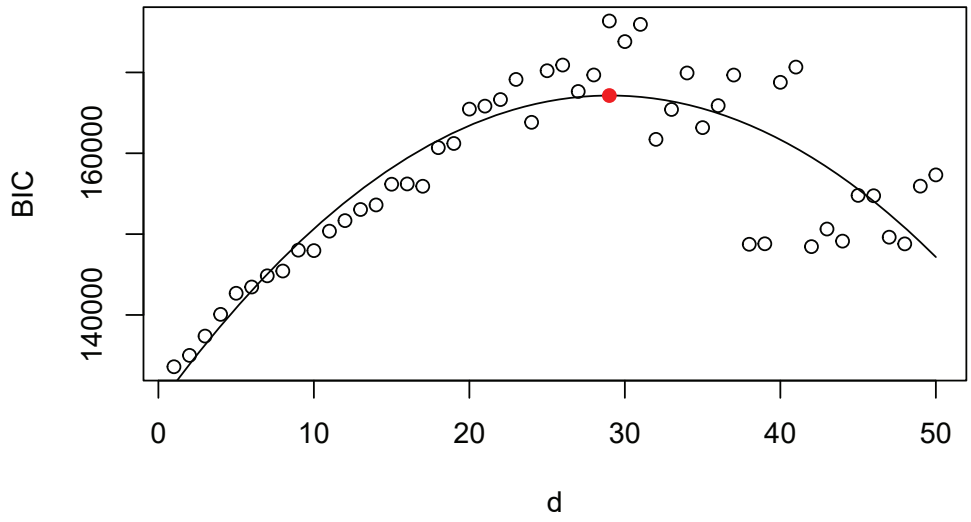


Figure 5.1: BIC values fitted by a quadratic regression model for the connectome on 1st subject and 1st scan. The hollow circles indicate the maximum BIC values across K for each fixed d. The curve shows the fitted quadratic regression model. The red solid dot shows the maximum of the curve, by which the model parameter d is estimated. In this case, $\hat{d} = 29$ is picked.

for each fixed d by

$$\text{BIC}(d) = \arg \max_{K \in [K_{\max}]} \text{BIC}(\hat{Z}; d, K) \tag{5.1}$$

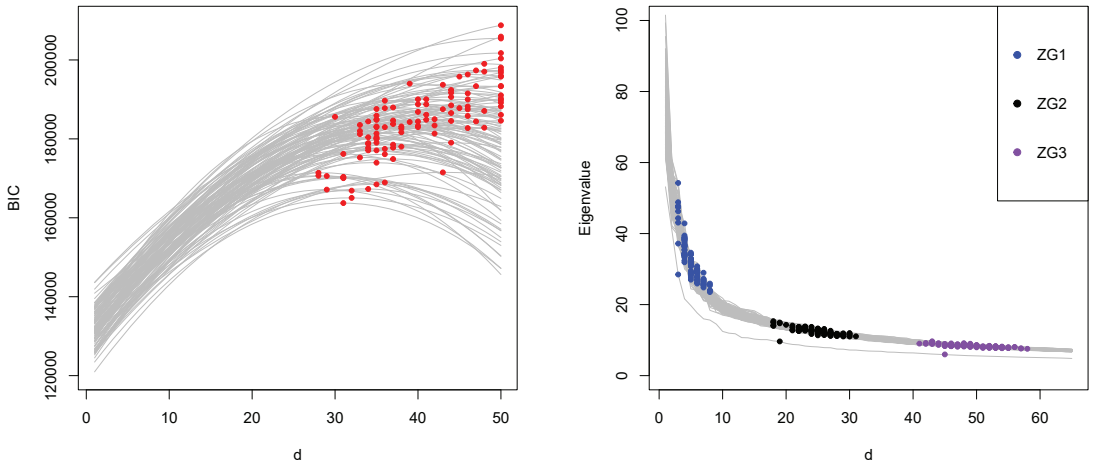
Then we fit a quadratic regression model on the sequence $(\text{BIC}(1), \ldots, \text{BIC}(d_{\max}))$. Finally, the estimator $\hat{d}$ is picked by maximizing the quadratic model over d. The estimator $\hat{K}$ is automatically the one that maximizes BIC with $d = \hat{d}$. As an example, we show the above procedure in figure 5.1. The hollow circles indicate the maximum BIC values $\text{BIC}(d)$ for each fixed d. The curve is the fitted quadratic regression model. $\hat{d} = 29$ is picked since it maximizes the regression

curve, shown by the red dot. Notice that this $\hat{d}$ coincides with the other one by maximizing BICs without regression.

5.1.3 Results of model selection

We first show the results of the estimates of latent position dimension $\hat{d}$ in Figure 5.2. As a comparison, we apply ZG method, described in Algorithm 2, on the same data set. In Figure 5.2(a) we draw the quadratic regression models by which BIC values (maximized over K) are fitted in gray curves for 114 graphs. The variation of estimating $\hat{d}$ by regression has been discussed in Section 5.1.2. The solid dots show the maximum points on each of the 114 curves, from which $\hat{d}$ is represented by the x-coordinate of the dots. Notice that for some graphs, $\hat{d}$ is picked on the boundary of the scanning range $d_{\max} = 50$. This is caused by the monotonicity of the regression model within the scanning range. Intuitively the estimates for those graphs could be larger if we extend the scanning range. In Figure 5.2(b), we draw the scree plots of eigenvalues of extended ASE for 114 graphs. Elbows are yielded by applying ZG method on the scree plot. We plot the first three elbows in solid dots in different colors, denoted by ZG1, ZG2 and ZG3 respectively. The x-coordinates of the dots are the corresponding estimate $\hat{d}$. Notice that the 1st, 2nd and 3rd elbows for all graphs are well separated. The $\hat{d}$ given by SMS is usually closer to the estimates given by ZG3.

Figure 5.3 present the estimates of the model parameter pair $(\hat{d}, \hat{K})$ for 114



(a) $\hat{d}$ picked by maximizing BICs via SMS (b) $\hat{d}$ picked by elbows of scree plot via ZG

Figure 5.2: The estimates of latent position dimension $\hat{d}$ picked by SMS and ZG. (a) The fitted BIC values (maximized over K) by quadratic regression are drawn in gray curves for 114 graphs. The estimate $\hat{d}$, indicated by the x-coordinate, is selected according to the maximum of each curve, which is shown by solid dots. (b) The scree plots are drawn in gray curves for 114 graphs. The estimate $\hat{d}$ is selected by determining the elbows, shown by solid dots in different colors for 1st, 2nd and 3rd elbows, via ZG algorithm.

connectomes. The red dots represent the results by simultaneous model selection, and others are the results by BIC $\circ$ ZG. The method BIC $\circ$ ZG is described in Algorithm 5, where ZG is applied on eigenvalues to get $\hat{d}$ first and BIC is applied on ASE with $\hat{d}$ dimension to get $\hat{K}$ after. Consequently, there are four points for each graph, representing the pair of estimates by SMS, BIC $\circ$ ZG1, BIC $\circ$ ZG2 and BIC $\circ$ ZG3 respectively. The coordinates of the points are slightly perturbed so as to view the occlusion. We observe that the $\hat{K}$ estimated by BIC $\circ$ ZG methods are spread out up to the boundary $K_{\max} = 30$. In contrast,

our SMS method gives a smaller and more concentrated estimate of number of clusters.

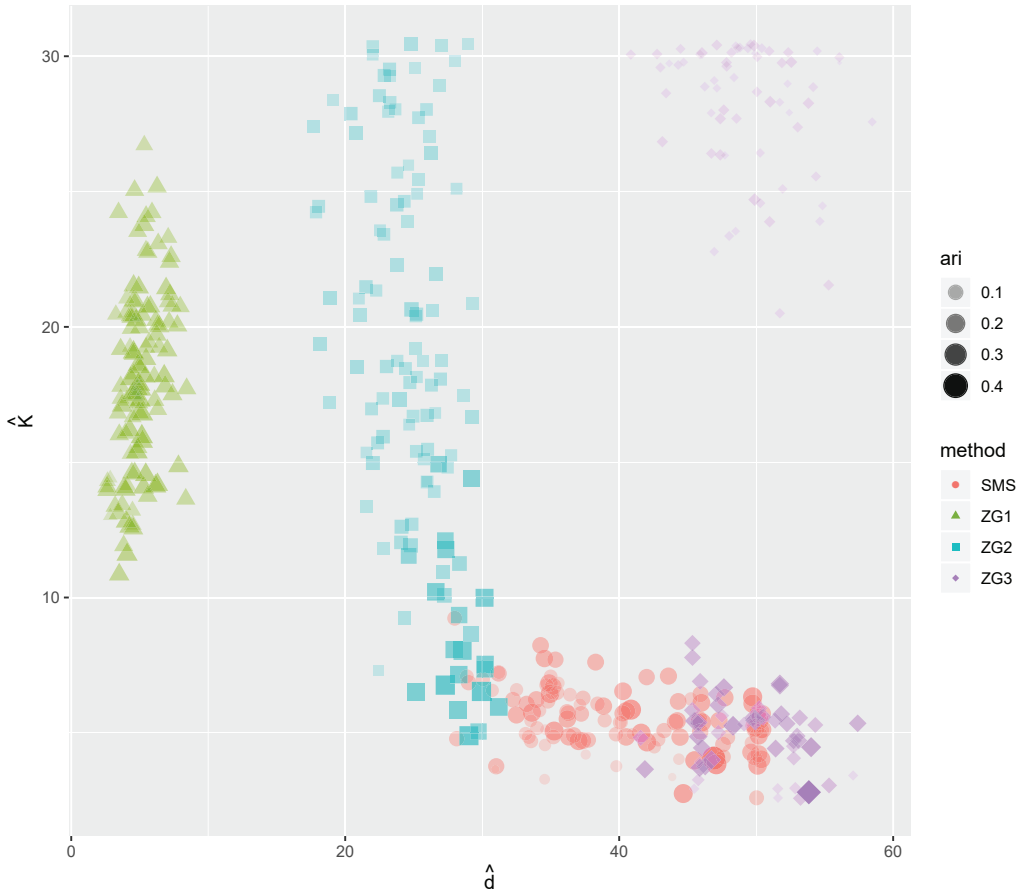


Figure 5.3: The estimates of the model parameter $(\hat{d}, \hat{K})$ for 114 connectomes. $D = 100$ and $K_{\max} = 30$. For each graph, four estimates by different methods are presented in different colors. The ARI values are indicated by the sizes of the dots. The coordinates of the points are slightly perturbed so as to view the occlusion. While $\hat{K}$ by BIC $\circ$ ZG methods are spread out, our SMS method gives a smaller and more concentrated estimate of number of clusters.

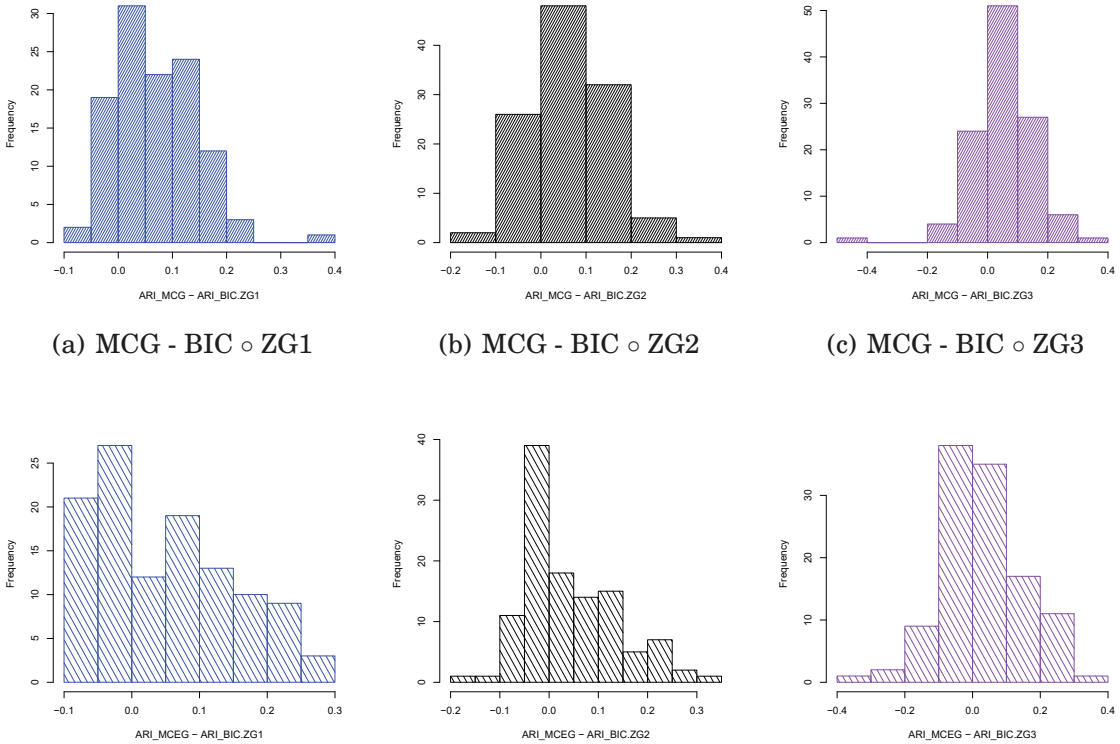
5.1.4 Results of clustering

We apply clustering methods on the data set to compare our MCG/MCEG algorithms via SMS framework, described in Algorithm 7 and Algorithm 8, with BIC ∘ ZG algorithm via CMS framework, described in Algorithm 5. We again use the adjusted rand index (ARI) as our cluster assessment criterion. The ARI is calculated by comparing the clustering results with three separate ground truths, namely hemisphere, tissue and the combination of the two. Specifically, each vertex of a connectome is assigned a label of left or right from the 2-cluster attribute hemisphere, and a label of gray or white from the 2-cluster attribute tissue. We also assign a label (left-gray, left-white, right-gray or right-white) from the 4-cluster attribute by combining the hemisphere and tissue.

For each graph and one specific algorithm, we have three ARIs indicating the clustering accuracy for three different attributes. We are interested in how well our MCG/MCEG algorithms perform compared with the traditional BIC ∘ ZG algorithms. As an example, Figure 5.4 shows the result of the paired difference of ARIs between MCG/MCEG and BIC ∘ ZG methods. Attribute hemisphere (left or right) is considered when computing ARI. Fixing two algorithms in competition, the differences of ARI are taken pair-wise for all 114 graphs. We plot the histogram of those differences. More positive values in the histogram indicates stronger evidence that MCG/MCEG outperforms BIC ∘ ZGs, since higher ARI indicates that clustering result is closer to the ground

truth. By Figure 5.4(a)-(c) we claim that MCG dominates all BIC $\circ$ ZGs, following the observation that obviously more difference values are positive. In Figure 5.4(d)-(f), although the number of positive values is close to that of negative ones, MCEG still wins to BIC $\circ$ ZGs slightly because of higher ARIs on average. Table 5.1 gives the results on all three attributes, where the number of graphs (out of 114) on which ARI of MCG/MCEG is strictly larger than existing methods is reported in the column "#win". Here we also consider the Louvain and Walktrap methods in the competition. We calculate a p-value by conducting a binomial test: $H_0 : p \leq 0.5$, $H_1 : p > 0.5$, where p is the probability that MCG/MCEG wins. The p-value evaluates the confidence of whether our method performs better in the connectome data set. The results show that MCG dominates in all cases against BIC $\circ$ ZGs. In addition, MCG/MCEG outperforms all other methods with respect to tissue attribute. Notice that the Louvain method demonstrates good performance for hemisphere and 4-block attributes, but it almost does not work (with very little ARIs in magnitudes) for tissue attribute. An analogous "two-truths" phenomenon has been discovered in the work of [85], where the authors find that Laplacian spectral embedding (LSE) better captures the hemisphere affinity structure while ASE better captures the tissue core-periphery structure. So in this manner Louvain is good at detecting the hemisphere affinity structure but is bad at detecting the tissue core-periphery structure. This is similar to the behavior of LSE on connectome

clustering [85].



(a) MCG - BIC ∘ ZG1 (b) MCG - BIC ∘ ZG2 (c) MCG - BIC ∘ ZG3

(d) MCEG - BIC ∘ ZG1 (e) MCEG - BIC ∘ ZG2 (f) MCEG - BIC ∘ ZG3

Figure 5.4: Illustration of the difference of ARI between MCG/MCEG and BIC ∘ ZG methods. The ARIs are computed by the ground truth in attribute hemisphere, with label left or right. The differences are taken pair-wise for all 114 graphs. (a)-(c) show the histogram of the 114 differences between MCG and BIC ∘ ZGs, while (d)-(f) show the histogram of those between MCEG and BIC ∘ ZGs. More positive values in the histogram indicates stronger evidence that MCG/MCEG outperforms BIC ∘ ZGs. While MCG dominates in all cases, MCEG wins over BIC ∘ ZGs slightly with higher ARI on average.

		MCG		MCEG	
		#win	p-value	#win	p-value
Hemisphere	BIC ∘ ZG1	93	5.8e-13	66	0.037
	BIC ∘ ZG2	86	7.6e-9	62	0.151
	BIC ∘ ZG3	85	2.4e-8	64	0.080
	Louvain	20	1	22	1
	Walktrap	70	5.6e-3	52	0.800
Tissue	BIC ∘ ZG1	69	0.009	41	0.998
	BIC ∘ ZG2	102	0	81	1.6e-6
	BIC ∘ ZG3	82	5.9e-7	56	0.537
	Louvain	110	0	101	0
	Walktrap	114	0	112	0
4-block	BIC ∘ ZG1	82	5.9e-7	57	0.463
	BIC ∘ ZG2	89	1.8e-10	60	0.256
	BIC ∘ ZG3	86	7.6e-9	67	0.024
	Louvain	32	1	34	1
	Walktrap	59	0.320	94	1.2e-13

Table 5.1: The evidence that MCG/MCEG outperforms BIC ∘ ZG in terms of ARI, which is evaluated by three different ground truths respectively: hemisphere, tissue and the combination of the two (4-block). The number of graphs (out of 114) on which ARI of MCG/MCEG is strictly larger than that of existing methods is reported in the column "#win". The p-value for a binomial test: $H_0 : p \leq 0.5$, $H_1 : p > 0.5$, where p is the probability that MCG/MCEG wins, is reported next to the corresponding number. The results show that MCG dominates in all cases against BIC ∘ ZGs. In addition, MCG/MCEG outperforms all other methods with respect to tissue attribute.

5.2 Larval Drosophila mushroom body connectome

5.2.1 Data description

In addition to the macro-scale connectomes discussed in section 5.1, there are also micro-scale connectomes generated from the raw data via electron microscopy. In these connectomes, vertices are the neurons and edges are the synapses between them. In this section we consider the data set of micro-scale connectomes characterizing the mushroom body (MB) in the larval Drosophila brain [24]. The data is collected by serial section transmission electron microscopy of the nervous system of a larval Drosophila. The MB connectome consists of 213 neurons, which are categorized into four distinct types, namely Kenyon cells (KC), Input Neurons (MBIN), Output Neurons (MBON) and Projection Neruons (PN). In contrast with the undirected graph of the macro-scale connectome obtained by diffusion MRI, the MB connectomes are directed graphs, with directed connectivity structure between certain pairs of neuron types. The possible connectivity directions are shown in figure 5.5. Therefore, this connectome can be modeled as a graph with a directed stochastic block model.

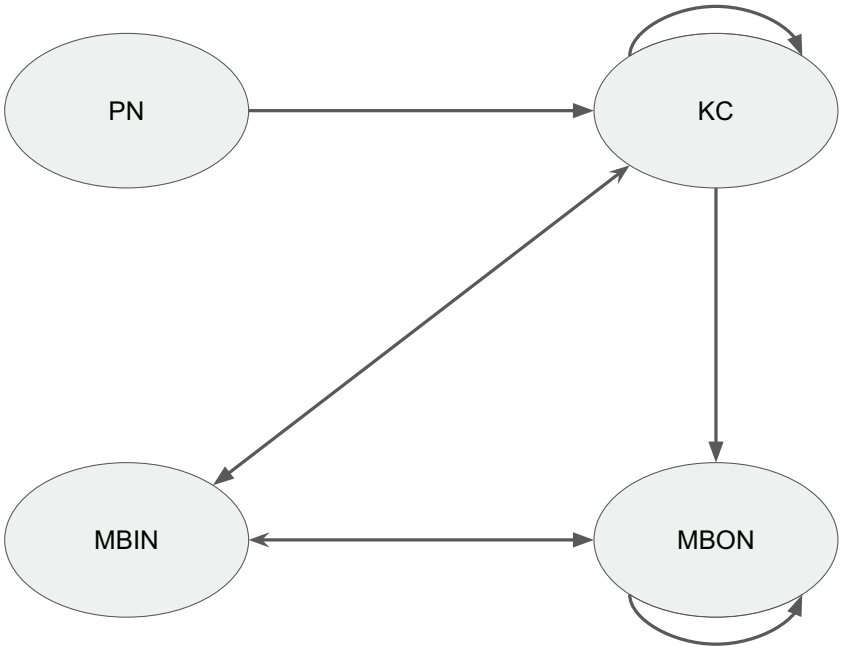


Figure 5.5: Illustration of the possible connectivity direction among four types of neurons in the larval Drosophila mushroom body connectome.

5.2.2 A model for directed graphs

Since the data of interest is now a directed graph, the corresponding adjacency matrix A is not symmetric (and thus is not positive semi-definite). So the extended adjacency spectral embedding for undirected graphs, described in Section 3.2.1, is not applicable. In order to use the spectral method, a directed version of adjacency spectral embedding is used [84]. The definition of directed ASE is as follows.

Definition 7 (Directed adjacency spectral embedding (directed ASE))**.** Let G be an directed graph of interest with n vertices, and $A \in \mathbb{R}^{n \times n}$ be its adjacency matrix. Let the singular value decomposition of A be

$$A = \hat{U}\hat{S}\hat{V}^T \tag{5.2}$$

Here, $\hat{S} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with singular values of A on its diagonal in descending order. That is, $\hat{S} = \text{diag}(\hat{\sigma}_1, \ldots, \hat{\sigma}_n)$ with $\hat{\sigma}_1 \geq \cdots \geq \hat{\sigma}_n$. $\hat{U}$ and $\hat{V}$ are orthogonal matrices whose columns are corresponding left and right singular vectors of A. For a given *embedding dimension* d satisfying $1 \leq d \leq n$, the *directed adjacency spectral embedding* (directed ASE) of G with dimension d is given by

$$\hat{X} = \left[\hat{U}_{[d]}\hat{S}_{[d]}^{\frac{1}{2}} \middle| \hat{V}_{[d]}\hat{S}_{[d]}^{\frac{1}{2}}\right] \tag{5.3}$$

where $\hat{U}_{[d]} \in \mathbb{R}^{n \times d}$ and $\hat{V}_{[d]} \in \mathbb{R}^{n \times d}$ is the submatrices of $\hat{U}$ and $\hat{V}$ with their first d columns respectively, and $\hat{S}_{[d]} \in \mathbb{R}^{d \times d}$ is the submatrix of $\hat{S}$ with its first d rows and columns, i.e. the diagonal matrix consisting of top d singular values of A.

The directed ASE $\hat{X} \in \mathbb{R}^{n \times 2d}$ is an intuitive variation of its undirected version. The central limit theorem in [2] suggests that if A is generated from a

K-block stochastic block model, rows of $\hat{X}$ behave approximately as a Gaussian mixture model with K mixture components. So the traditional spectral methods, for example BIC $\circ$ ZG, can be applied on $\hat{X}$ for clustering purposes. As always we are facing the model selection problem, just like in the undirected version. Similar to (3.11), we define the extended directed ASE to be

$$\hat{Z} = \left[\hat{U}_{[D]}\hat{S}_{[D]}^{\frac{1}{2}} \middle| \hat{V}_{[D]}\hat{S}_{[D]}^{\frac{1}{2}}\right] \tag{5.4}$$

where D is a constant, usually picked as a loose upper bound of the true embedding dimension d. Within the framework of simultaneous model selection, we seek a model that describes the extended directed ASE. For convenience, let $\tilde{Z} \in \mathbb{R}^{n\times 2D}$ be a matrix by permuting the columns of $\hat{Z}$ so that the $(2i-1)$-th and $(2i)$-th columns of $\tilde{Z}$ are the i-th and $(D+i)$-th column of $\hat{Z}$ respectively, for $i = 1, \ldots, D$. That is, the top normalized left and right singular vectors of A are concatenated alternately in $\tilde{Z}$. Analogous to model 1, we propose a model for $\tilde{Z}$ as follows:

Model 2 (GMM for extended ASE of directed graphs)**.** *Let*

$$f(\cdot;\theta(d,K)) = \sum_{k=1}^{K} \pi^{(k)}\varphi(\cdot;\mu^{(k)},\Sigma^{(k)}) \tag{5.5}$$

be a family of density functions for a $2D$ dimensional GMM random vector, where $\{\pi^{(k)}\}_{k=1}^{K}$ are the mixing probabilities, $\{\mu^{(k)}\}_{k=1}^{K}$ are the mean vectors, and

$\{\Sigma^{(k)}\}_{k=1}^{K}$ *are the covariance matrices. Furthermore, they satisfy*

$$\sum_{k=1}^{K} \pi^{(k)} = 1 \tag{5.6}$$

$$\mu^{(k)} = [\mu_1^{(k)}, \ldots, \mu_{2d}^{(k)}, 0, \ldots, 0]^T \tag{5.7}$$

and

$$\Sigma^{(k)} = \begin{bmatrix} \tilde{\Sigma}^{(k)} & 0 \\ 0 & \Sigma_2^{(k)} \end{bmatrix} \tag{5.8}$$

where $\tilde{\Sigma}^{(k)}$ *is a* $2d \times 2d$ *positive semidefinite matrix, and* $\Sigma_2^{(k)}$ *is a* $(2D-2d) \times (2D-2d)$ *block diagonal matrix whose diagonals are* $(D-d)$ *identical* 2×2 *matrices. In this notation,* $\theta(d, K)$ *denotes the parameters* $\{\rho^{(k)}, \mu^{(k)}, \Sigma^{(k)}\}_{k=1}^{K}$*, which belong to the parameter space* $\Theta(d, K)$*.*

The intuition of this model is that the informative part of $\tilde{Z}$ and redundant part of $\tilde{Z}$ are separated by a model parameter d, where the rows of the informative part follow a standard $2d$-dimensional GMM, and while the rows of the redundant part follow a $2(D-d)$-dimensional GMM with mean zero and block diagonal covariance matrices. The 2×2 block structure is based on the belief that the j-th left and right singular vectors are correlated. Consequently, our conjecture states, for any $i \in [n]$,

$$\tilde{Z}_i \sim f(\cdot; \theta^*(d_0, K_0)) \tag{5.9}$$

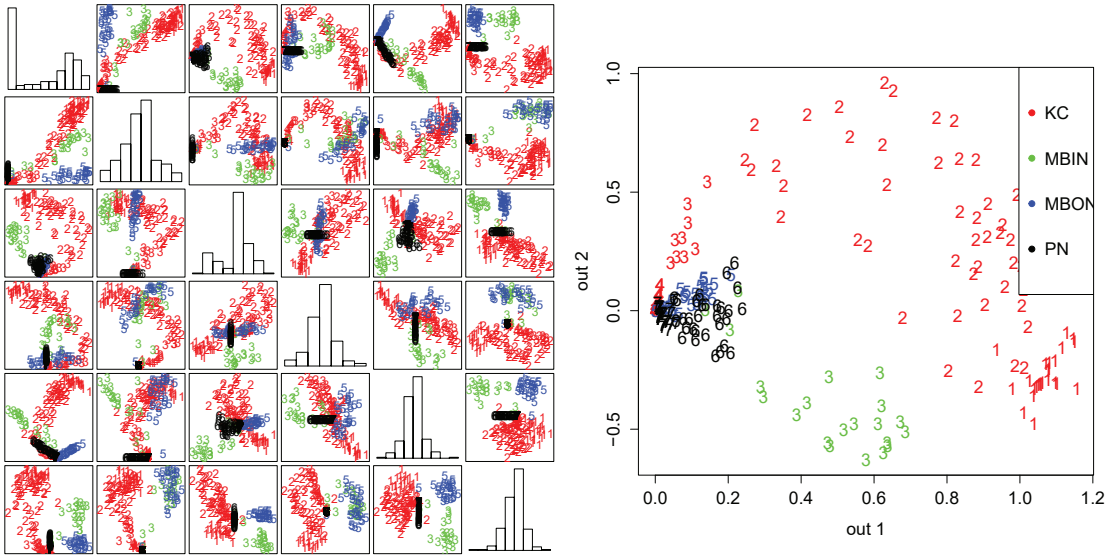
approximately for sufficiently large n, where $f(\cdot;\theta(d,K))$ is the density function defined in model 2, d_0 is the true dimension of latent position, K_0 is the true number of blocks, and $\theta^*(d,K)$ is the true underlying parameter of the GMM.

With this model we can now apply the simultaneous model selection, described in Algorithm 6, as well as the MCG/MCEG algorithms, described in Algorithm 7 and Algorithm 8, on directed graphs by substituting the directed version of extended ASE $\tilde{Z}$ and model 2.

5.2.3 Clustering analysis

The graph of the MB connectome consists of 213 vertices, and each vertex is assigned a label from {KC, MBIN, MBON, PN} indicating its type of neurons. We set $D = 20$ to be the dimension of the extended directed ASE, yielding $\tilde{Z} \in \mathbb{R}^{213\times 40}$. We apply our MCG algorithm via SMS with model 2 on $\tilde{Z}$. The estimates of model parameters are respectively $\hat{d} = 3$ and $\hat{k} = 7$. The clustering results are depicted by the pair plot in Figure 5.6(a). The pair plot shows the first $2\hat{d} = 6$ dimensions of the extended directed ASE $\tilde{Z}$. The neuron type of each vertex is shown by the color, while the label of clustering result is shown by the digit. For ease of illustration, we zoom in on the subplot corresponding to the 1st and 2nd left singular vectors in Figure 5.6(b). We can see a clear clustering result by which the vertices are separated according to the neuron types. The only concern is clustering label 3, which mixes some of the KC

and MBIN neurons. This can be seen more clearly in Table 5.2, where we have reported the number of vertices of each neuron type in each cluster. We observe that, except for cluster 3, the clustering works almost perfectly in the sense that one cluster contains just one neuron type. The adjusted rand index of this result is 0.468.



(a) Pair plot for the extended directed ASE

(b) A 2-dimensional visualization on the 1st and 2nd left singular vectors

Figure 5.6: (a) A pair plot for the extended directed ASE $\tilde{Z}$ on the first 6 dimensions. MCG is applied on $\tilde{Z}$, yielding $\hat{d} = 3$ and $\hat{k} = 7$. The neuron type of each vertex is shown by the color, while the label of clustering result is shown by the digit. (b) A 2-dimensional visualization on the 1st and 2nd left singular vectors for ease of illustration.

We compare the performance of MCG/MCEG algorithms with BIC $\circ$ ZG. We report the estimates of $\hat{d}$, $\hat{K}$ and the ARI for different methods in table 5.3. Although MCG has the least ARI (which may be caused by the difference of GMM), the estimation of $\hat{d}$ and $\hat{K}$ given by simultaneous model selection

	1	2	3	4	5	6	7
KC	25	48	9	16	0	0	2
MBIN	0	0	21	0	0	0	0
MBON	0	0	0	0	29	0	0
PN	0	0	0	0	0	30	33

Table 5.2: The number of vertices of each neuron type in each of the $\hat{K} = 7$ clusters. Clustering is conducted by MCG algorithm with model 2. The clustering works well except for cluster 3 in the sense that one cluster contains a single neuron type. The ARI is 0.468.

	$\hat{d}$	$\hat{K}$	ARI
MCG	3	7	0.468
MCEG	3	7	0.574
BIC $\circ$ ZG1	1	4	0.621
BIC $\circ$ ZG2	3	7	0.574
BIC $\circ$ ZG3	4	2	0.481

Table 5.3: The estimates of $\hat{d}$, $\hat{K}$ and the ARI for different methods.

(SMS) coincides with the one given by BIC $\circ$ ZG2, and they may be the best estimates of $\hat{d}$ and $\hat{K}$ among all algorithms. To see this, we show the scree plot of the singular values of the adjacency matrix A in Figure 5.7. The 1st, 2nd and 3rd elbows $\hat{d}$ equal to $1, 3, 4$ respectively by ZG algorithms. We notice that an obvious cut-off of the scree plot is at $\hat{d} = 3$. This is evidence of the good estimation of $\hat{d}$ by SMS. We also show the ARI values of the clustering results by GMM $\circ$ ASE given the embedding dimension $\hat{d}$ and mixture complexity $\hat{K}$,

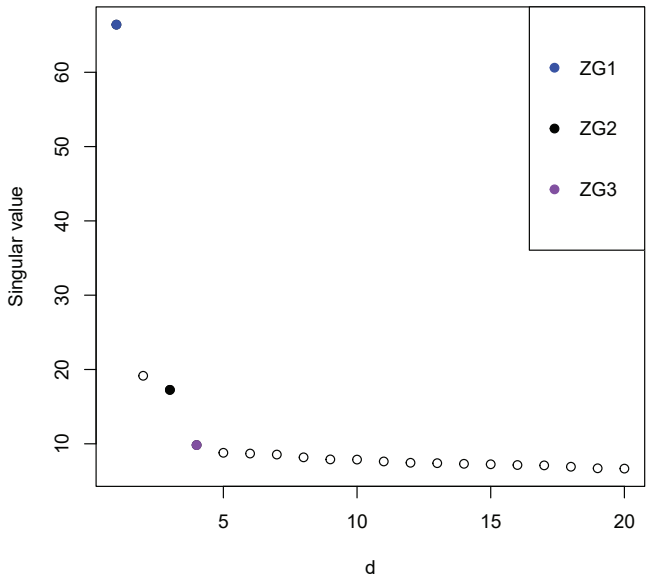


Figure 5.7: The scree plot of the singular values of the adjacency matrix A. The 1st, 2nd and 3rd elbows picked by ZG1, ZG2 and ZG3 respectively are shown by solid dots. Intuitively, the cut-off of the scree plot is at $\hat{d} = 3$.

in Table 5.4. We observe that the best ARI is achieved when $\hat{d} = 3$ and $\hat{K} = 6$. So by Table 5.3, the $(\hat{d}, \hat{K})$ given by MCG, MCEG and BIC $\circ$ ZG2 are closest to the optimal estimation. On the other hand, BIC $\circ$ ZG1 gives $\hat{d} = 1$ which may be too small to contain all the clustering informative (see the scree plot), while BIC $\circ$ ZG3 gives $\hat{K} = 2$, which may be too small considering the number of neuron types. So the $(\hat{d}, \hat{K})$ given by MCG, MCEG and BIC $\circ$ ZG2 may be superior, although BIC $\circ$ ZG1 has the highest ARI (which may be caused by the coincidence of number of clusters). Moreover, as we mentioned in Section 4.1.1, determining which elbow should be used in the ZG method is subjective. It is also worth mentioning that the results of BIC algorithm variate in a different

$\hat{K}$ \ $\hat{d}$	1	2	3	4	5
2	0.508	0.508	0.508	0.481	0.498
3	0.474	0.404	0.459	0.339	0.403
4	0.621	0.368	0.630	0.249	0.524
5	0.610	0.575	0.624	0.430	0.489
6	0.607	0.623	**0.671**	0.525	0.594
7	0.583	0.598	0.574	0.443	0.463
8	0.557	0.520	0.541	0.389	0.436

Table 5.4: The ARI of the clustering results by GMM $\circ$ ASE given the embedding dimension $\hat{d}$ and mixture complexity $\hat{K}$.

version of Mclust [88]. In an early version of Mclust, the highest ARI is given by BIC $\circ$ ZG2, with $\hat{d} = 3$ and $\hat{K} = 6$ (see [84]). So we claim that the simultaneous model selection estimates the model parameters in a robust way.

Chapter 6

Conclusion

This thesis attempts to address the issue of model selection for spectral vertex clustering by establishing a novel model selection framework specifically for vertex clustering on graphs with stochastic block model.

In the first part of the thesis we propose the extended adjacency spectral embedding (extended ASE) in which the embedding is performed with a fixed dimension. Under the framework of model-based clustering, we propose a family of specific Gaussian mixture models (GMM) to parameterize the entire extended ASE. The basis of the model is comprised of a state-of-the-art distributional result for the informative dimensions, as well as a strong evidence of principled simulations for redundant dimensions.

In the second part of the thesis, we propose a simultaneous model selection (SMS) framework to address the issue occurring in the consecutive model se-

lection. The framework is specifically tailored for a vertex clustering task on the graph with stochastic block model. In contrast with consecutive model selection, our SMS identifies the embedding dimension, mixture complexity and membership of each vertex simultaneously. Moreover, we state and prove a theorem on the consistency of model parameter estimates. The theorem claims that the estimates in the model selection procedure given by our SMS method converge to the underlying truth for the large size of the graph, provided the extended ASE follows the distribution in our proposed model. Based on SMS, we also develop two heuristic algorithms to solve the vertex clustering problems. The effectiveness of the algorithms are verified in the simulations.

The third part of the thesis is a demonstration of our methodology on real data sets of connectomes, the kinds of graphs representing the neuronal connectivity in brains. We explain the variety of our algorithms in certain scenarios, such as in the case of noisy data and directed graphs. Finally, the results successfully interpret the structural attributes of the connectomes.

Bibliography

[1] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer, 1998.

[2] Avanti Athreya, Carey E Priebe, Minh Tang, Vince Lyzinski, David J Marchette, and Daniel L Sussman. A limit theorem for scaled eigenvectors of random dot product graphs. *Sankhya A*, 78(1):1–18, 2016.

[3] David J Bartholomew and Martin Knott. *Latent variable models and factor analysis*, volume 7. Arnold London, 1999.

[4] Maurice S Bartlett. A note on the multiplying factors for various χ 2 approximations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 296–298, 1954.

[5] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[6] JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West. Bayesian clustering with variable and transformation selections. In *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*, volume 249. Oxford University Press, USA, 2003.

[7] Peter J Bickel and Aiyou Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, pages pnas–0907096106, 2009.

[8] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725, 2000.

[9] Horst Bischof, Aleš Leonardis, and Alexander Selb. Mdl principle for robust vector quantisation. *Pattern Analysis & Applications*, 2(1):59–72, 1999.

[10] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[11] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

[12] Charles Bouveyron and Camille Brunet-Saumard. Discriminative variable selection for clustering with the sparse fisher-em algorithm. *Computational Statistics*, 29(3-4):489–513, 2014.

[13] Edward T Bullmore and Danielle S Bassett. Brain graphs: graphical models of the human brain connectome. *Annual review of clinical psychology*, 7:113–140, 2011.

[14] Jonathan G Campbell, Chris Fraley, D Stanford, Fionn Murtagh, and Adrian E Raftery. Model-based methods for textile fault detection. *International Journal of Imaging Systems and Technology*, 10(4):339–346, 1999.

[15] Gilles Celeux, Marie-Laure Martin-Magniette, Cathy Maugis-Rabusseau, and Adrian E Raftery. Comparing model selection and regularization approaches to variable selection in model-based clustering. *Journal de la Societe francaise de statistique (2009)*, 155(2):57, 2014.

[16] Gilles Celeux and Gilda Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of classification*, 13(2):195–212, 1996.

[17] Wei-Chien Chang. On using principal components before separating a

mixture of two multivariate normal distributions. *Applied Statistics*, pages 267–275, 1983.

[18] Sourav Chatterjee et al. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

[19] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[20] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[21] Abhijit Dasgupta and Adrian E Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93(441):294–302, 1998.

[22] Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[23] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[24] K Eichler, F Li, A Litwin-Kumar, Y Park, I Andrade, CM Schneider-Mizell, T Saumweber, A Huser, D Bonnery, B Gerber, et al. The complete wiring diagram of a high-order learning and memory center, the insect mushroom body. *Nature*, 548(175-182):23, 2017.

[25] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[26] Brian S Everitt. Mixture distributionsi. *Encyclopedia of statistical sciences*, 7, 2004.

[27] Donniell E Fishkind, Daniel L Sussman, Minh Tang, Joshua T Vogelstein, and Carey E Priebe. Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown. *SIAM Journal on Matrix Analysis and Applications*, 34(1):23–39, 2013.

[28] Michael Fop and Thomas Brendan Murphy. Variable selection methods for model-based clustering. *Statistics Surveys*, 12:18–65, 2018.

[29] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631, 2002.

[30] Serge Frontier. Étude de la décroissance des valeurs propres dans une analyse en composantes principales: Comparaison avec le moddle

du bâton brisé. *Journal of experimental marine Biology and Ecology*, 25(1):67–75, 1976.

[31] Giuliano Galimberti, Annamaria Manisi, and Gabriele Soffritti. Modelling the role of variables in model-based cluster analysis. *Statistics and Computing*, 28(1):145–169, 2018.

[32] Giuliano Galimberti, Angela Montanari, and Cinzia Viroli. Penalized factor mixture analysis for variable selection in clustered data. *Computational Statistics & Data Analysis*, 53(12):4301–4310, 2009.

[33] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[34] Louis Guttman. Some necessary conditions for common-factor analysis. *Psychometrika*, 19(2):149–161, 1954.

[35] Patric Hagmann. From diffusion mri to brain connectomics. 2005.

[36] Greg Hamerly and Charles Elkan. Learning the k in k-means. In *Advances in neural information processing systems*, pages 281–288, 2004.

[37] André Hardy. On the number of clusters. *Computational Statistics & Data Analysis*, 23(1):83–96, 1996.

[38] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

[39] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[40] Jun Huan, Wei Wang, and Jan Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *null*, page 549. IEEE, 2003.

[41] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[42] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

[43] Donald A Jackson. Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. *Ecology*, 74(8):2204–2214, 1993.

[44] Anil K Jain, Robert PW Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.

[45] Chuntao Jiang, Frans Coenen, and Michele Zito. A survey of fre-

quent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(1):75–105, 2013.

[46] George H John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier, 1994.

[47] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[48] Tosio Kato. On the upper and lower bounds of eigenvalues. *Journal of the Physical Society of Japan*, 4(4-6):334–339, 1949.

[49] Christine Keribin. Consistent estimation of the order of mixture models. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 49–66, 2000.

[50] Gregory Kiar, Eric Bridgeford, Will Gray Roncal, Vikram Chandrashekhar, Disa Mhembere, Sephira Ryman, Xi-Nian Zuo, Daniel S Marguiles, R Cameron Craddock, Carey E Priebe, et al. A high-throughput pipeline identifies robust connectomes but troublesome variability. *bioRxiv*, page 188706, 2018.

[51] YeongSeog Kim, W Nick Street, and Filippo Menczer. Feature selection in unsupervised learning via evolutionary search. In *Proceedings*

of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 365–369. ACM, 2000.

[52] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[53] Daphne Koller and Mehran Sahami. Toward optimal feature selection. Technical report, Stanford InfoLab, 1996.

[54] Martin HC Law, Mario AT Figueiredo, and Anil K Jain. Simultaneous feature selection and clustering using mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1154–1166, 2004.

[55] DN Lawley. On testing a set of correlation coefficients for equality. *The Annals of Mathematical Statistics*, 34(1):149–151, 1963.

[56] David Lazer, Alex Sandy Pentland, Lada Adamic, Sinan Aral, Albert Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

[57] Can M Le, Elizaveta Levina, and Roman Vershynin. Concentration and

regularization of random graphs. *Random Structures & Algorithms*, 51(3):538–561, 2017.

[58] Jing Lei, Alessandro Rinaldo, et al. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

[59] Geng Li, Murat Semerci, Bulent Yener, and Mohammed J Zaki. Graph classification via topological and label attributes. In *Proceedings of the 9th international workshop on mining and learning with graphs (MLG), San Diego, USA*, volume 2, 2011.

[60] Huan Liu and Hiroshi Motoda. *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media, 2012.

[61] Huan Liu, Hiroshi Motoda, and Lei Yu. Feature selection with selective sampling. In *ICML*, pages 395–402, 2002.

[62] Huan Liu, Rudy Setiono, et al. A probabilistic approach to feature selection-a filter solution. In *ICML*, volume 96, pages 319–327. Citeseer, 1996.

[63] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[64] Linyuan Lu and Xing Peng. Spectra of edge-independent random graphs. *arXiv preprint arXiv:1204.6207*, 2012.

[65] Vince Lyzinski, Daniel L Sussman, Minh Tang, Avanti Athreya, Carey E Priebe, et al. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electronic Journal of Statistics*, 8(2):2905–2922, 2014.

[66] Vince Lyzinski, Minh Tang, Avanti Athreya, Youngser Park, and Carey E Priebe. Community detection and classification in hierarchical stochastic blockmodels. *IEEE Transactions on Network Science and Engineering*, 4(1):13–26, 2017.

[67] Matthieu Marbac and Mohammed Sedki. Variable selection for model-based clustering using the integrated complete-data likelihood. *Statistics and Computing*, 27(4):1049–1063, 2017.

[68] Cathy Maugis, Gilles Celeux, and M-L Martin-Magniette. Variable selection in model-based clustering: A general variable role modeling. *Computational Statistics & Data Analysis*, 53(11):3872–3882, 2009.

[69] Cathy Maugis, Gilles Celeux, and Marie-Laure Martin-Magniette. Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3):701–709, 2009.

[70] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[71] Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 84. Marcel Dekker, 1988.

[72] Paul D McNicholas. Model-based clustering. *Journal of Classification*, 33(3):331–373, 2016.

[73] Marina Meilă. Comparing clusteringsan information based distance. *Journal of multivariate analysis*, 98(5):873–895, 2007.

[74] Volodymyr Melnykov, Ranjan Maitra, et al. Finite mixture models and model-based clustering. *Statistics Surveys*, 4:80–116, 2010.

[75] Glenn W Milligan and Martha C Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[76] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[77] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[78] Roberto Imbuzeiro Oliveira. Concentration of the adjacency matrix and of the laplacian in random graphs with independent edges. *arXiv preprint arXiv:0911.0600*, 2009.

[79] Wei Pan and Xiaotong Shen. Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8(May):1145–1164, 2007.

[80] Youngser Park, Carey E Priebe, and Abdou Youssef. Anomaly detection in time series of graphs using fusion of graph invariants. *IEEE Journal on Selected Topics in Signal Processing*, 7(1):67–75, 2013.

[81] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.

[82] Richard A Pimentel. *Morphometrics, the multivariate analysis of biological data*. Kendall/Hunt Pub. Co., 1979.

[83] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *International symposium on computer and information sciences*, pages 284–293. Springer, 2005.

[84] Carey E Priebe, Youngser Park, Minh Tang, Avanti Athreya, Vince Lyzinski, Joshua T Vogelstein, Yichen Qin, Ben Cocanougher, Katharina Eichler, Marta Zlatic, et al. Semiparametric spectral modeling of the drosophila connectome. *arXiv preprint arXiv:1705.03297*, 2017.

[85] Carey E Priebe, Youngser Park, Joshua T Vogelstein, John M Conroy,

Vince Lyzinskic, Minh Tang, Avanti Athreya, Joshua Cape, and Eric Bridgeford. On a'two truths' phenomenon in spectral graph clustering. *arXiv preprint arXiv:1808.07801*, 2018.

[86] Stephen R Proulx, Daniel EL Promislow, and Patrick C Phillips. Network thinking in ecology and evolution. *Trends in ecology & evolution*, 20(6):345–353, 2005.

[87] Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Advances in Neural Information Processing Systems*, pages 3120–3128, 2013.

[88] Adrian E Raftery and Nema Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006.

[89] Gunter Ritter. *Robust cluster analysis and variable selection*. Chapman and Hall/CRC, 2014.

[90] Kathryn Roeder and Larry Wasserman. Practical bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92(439):894–902, 1997.

[91] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and

the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[92] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[93] Patrick Rubin-Delanchy, Carey E Priebe, Minh Tang, and Joshua Cape. A statistical interpretation of spectral embedding: the generalised random dot product graph. *arXiv preprint arXiv:1709.05506*, 2017.

[94] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[95] Luca Scrucca. Genetic algorithms for subset selection in model-based clustering. In *Unsupervised Learning Algorithms*, pages 55–70. Springer, 2016.

[96] Padhraic Smyth. Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and computing*, 10(1):63–72, 2000.

[97] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: a structural description of the human brain. *PLoS computational biology*, 1(4):e42, 2005.

[98] Derek Stanford and Adrian E Raftery. Principal curve clustering with noise. Technical report, Citeseer, 1997.

[99] Douglas Steinley and Michael J Brusco. Selection of variables in cluster analysis: An empirical comparison of eight procedures. *Psychometrika*, 73(1):125, 2008.

[100] Catherine A Sugar and Gareth M James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763, 2003.

[101] Daniel L Sussman, Minh Tang, Donniell E Fishkind, and Carey E Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499):1119–1128, 2012.

[102] Daniel L Sussman, Minh Tang, and Carey E Priebe. Consistent latent position estimation and vertex classification for random dot product graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):48–57, 2014.

[103] Shakira Suwan, Dominic S Lee, Runze Tang, Daniel L Sussman, Minh Tang, Carey E Priebe, et al. Empirical bayes estimation for the stochastic blockmodel. *Electronic Journal of Statistics*, 10(1):761–782, 2016.

[104] Mahlet G Tadesse, Naijun Sha, and Marina Vannucci. Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association*, 100(470):602–617, 2005.

[105] Minh Tang, Avanti Athreya, Daniel L Sussman, Vince Lyzinski, Youngser Park, and Carey E Priebe. A semiparametric two-sample hypothesis testing problem for random graphs. *Journal of Computational and Graphical Statistics*, 26(2):344–354, 2017.

[106] Minh Tang, Carey E Priebe, et al. Limit theorems for eigenvectors of the normalized laplacian for random graphs. *The Annals of Statistics*, 46(5):2360–2415, 2018.

[107] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[108] Sijian Wang and Ji Zhu. Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*, 64(2):440–448, 2008.

[109] Michael D Ward, Katherine Stovel, and Audrey Sacks. Network analysis and political science. *Annual Review of Political Science*, 14:245–264, 2011.

[110] Halbert White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, pages 1–25, 1982.

[111] Benhuai Xie, Wei Pan, and Xiaotong Shen. Variable selection in penalized model-based clustering via regularization on grouped parameters. *Biometrics*, 64(3):921–930, 2008.

[112] Stephen J Young and Edward R Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer, 2007.

[113] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5(Oct):1205–1224, 2004.

[114] Kim E Zerba and James P Collins. Spatial heterogeneity and individual variation in diet of an aquatic top predator. *Ecology*, 73(1):268–279, 1992.

[115] Mu Zhu and Ali Ghodsi. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis*, 51(2):918–930, 2006.

Biographical Statement

Congyuan Yang was born in Shanghai, China in 1986. He received his Bachelor degree in Electrical Engineering from Tsinghua Univerisity in 2009. He received his M.S. degree in Electrical Engineering from Tsinghua University, where he worked on survivability and resource optimization of optical networks, in 2012. He started his doctoral research focusing on statistical learning at Department of Electrical and Computer Engineering at Johns Hopkins University in 2012.